

**Efficient Inversion of Multi-frequency and Multi-Source
Electromagnetic Data**

**Final Project Report
15 July 2002-14 July 2006**

Gary D. Egbert

College of Oceanic and Atmospheric Sciences
COAS Admin. Bldg. 104
Oregon State University
Corvallis OR, 97331-5503

Prepared for the
U.S. Department of Energy
Office of Basic Energy Sciences
Under Grant DE-FG0302ER15318

1 Project Objectives

The project focused on development of efficient but robust non-linear inversion algorithms for electromagnetic induction data, in particular for data collected with multiple receivers, and multiple transmitters (i.e., at multiple frequencies or with multiple source geometries). This situation is extremely common in geophysical EM subsurface imaging methods (but note that in some multi-transmitter problems, e.g., helicopter-borne electromagnetic data, there is essentially only one receiver for each transmitter; the methods we describe here are not particularly suited to this challenging problem). Our formulation of the inverse problem is fairly standard: we seek to construct models that fit observational data, and possibly other constraints, by minimizing a functional penalizing both data misfit, and deviations of the modeled conductivity from prior assumptions. Once the penalty functional is defined, inversion is essentially reduced to a problem in non-linear optimization. A focus of our research has been on development and testing more efficient approaches to solving this optimization problem, by taking advantage of some special features of multi-transmitter EM inverse problems.

A key observation is that for such multi-transmitter problems each step in commonly used linearized iterative limited memory search schemes such as conjugate gradients (CG) requires solution of forward and adjoint EM problems for each of the N_f frequencies or sources, essentially generating data sensitivities for an N_f dimensional data-subspace. These multiple sensitivities allow a good approximation to the full Jacobian of the data mapping to be built up in many fewer search steps than would be required by application of textbook optimization methods, which take no account of the multiplicity of forward problems that must be solved for each search step. We have applied this idea to develop a hybrid inversion scheme that combines features of the iterative limited memory type methods with a Newton-type approach using a partial calculation of the Jacobian. Initial tests on 2D problems show that the new approach produces results essentially identical to a Newton type Occam minimum structure inversion, while running more rapidly than an iterative (fixed regularization parameter) CG style inversion. Memory requirements, while greater than for something like CG, are modest enough that even in 3D the scheme should allow 3D inverse problems to be solved on a common desktop PC, at least for modest (~ 100 sites, 15-20 frequencies) data sets.

There are a number of alternatives to the specific approach we have developed for multi-transmitter problems, for example using formulations of the search algorithm in the data or model spaces. To more effectively test and compare a range of possible approaches we have developed 2D and 3D inversion software using a modular, object oriented, approach. Although originally envisaged as an efficient way to prototype the specialized inversion algorithms we are developing, in the course of this project it has become clear that there are many other justifications for this programming approach. For example, the modular system that we have developed cleanly separates the forward modeling code (the core of any inversion scheme), the optimization algorithms, conductivity parameterizations and regularization, and data functionals. Modifications and extensions (e.g., to consider new data types; or to parameterize the model differently) require modifications of only a few specific modules, with minimal or no impact on other modules. Solvers tailored to very different EM inverse problem (e.g., for active source methods widely used in shallow and commercial applications, or passive source methods such as magnetotellurics (MT)) can be built from essentially the

same pieces. New developments in modeling methodology (e.g., finite element schemes, now being developed for more efficient and accurate modeling of sea-floor EM data) could also in future be “plugged in” to our general framework, as most inversion methods operate in either the data or model space, and none of the operations in these spaces need depend on details of the forward mapping. We believe that the results of our efforts to develop modular 3D EM inversion software will prove as valuable as the new search schemes develop using this system.

2 Accomplishments

Accomplishments during this project are in three general areas, as detailed below, and in the attached preprint, which has been submitted to *Inverse Problems*.

2.1 Development of a 3D data space MT inversion

In collaboration with Dr. Weerachai Siripunvaraporn of Mahidol University in Bangkok, Thailand, we have developed 3D MT forward modeling and inversion codes. The forward modeling code (Siripunvaraporn et al., 2002) is based on solution of the 3D quasi-static frequency domain induction equations, formulated in terms of a second order system for the electric fields on a staggered grid. The inversion (Siripunvaraporn et al., 2005a) is a data space Gauss-Newton scheme, using a full calculation of the Jacobian. Although subsequent developments show that considerably more efficient schemes are possible, this initial work already demonstrates that a Gauss-Newton type inversion based on sensitivity calculations can be practical, if a dataspace approach is used. For modest model parameterizations (e.g., $N_x = 35, N_y = 35, N_z = 30$) and data sets (full impedances (8 real responses) for $N_s = 36$ sites, $N_f = 10$ frequencies) the code will run on a modern PC with sufficient memory (2 Gb) in roughly 1-2 days. We have applied the codes to studies with synthetic (Siripunvaraporn et al., 2005b) and real data (Siripunvaraporn et al., 2004). The source code has recently been made available for free non-commercial use at <http://mucc.mahidol.ac.th/~scwsp/wsinv3dmt/index.htm>, and use of the code has been cited in at least a half dozen other publications already.

This 3D inversion code was not developed using the modular approach that we discuss next, but it has provided a good starting point for further developments, giving us experience with all of the components needed for the modular code. These 3D codes, which have now been thoroughly tested and applied to a number of synthetic and real test cases, have also been very useful for debugging and validating the modular versions. The bulk of this work was done by Dr. Siripunvaraporn, in consultation with the PI, and other colleagues in Japan. We have exchanged visits, and maintained an active collaboration, resulting in 4 joint publications over the project period (Siripunvaraporn et al., 2002; 2004, 2005a,b), plus a number of presentations at international meetings.

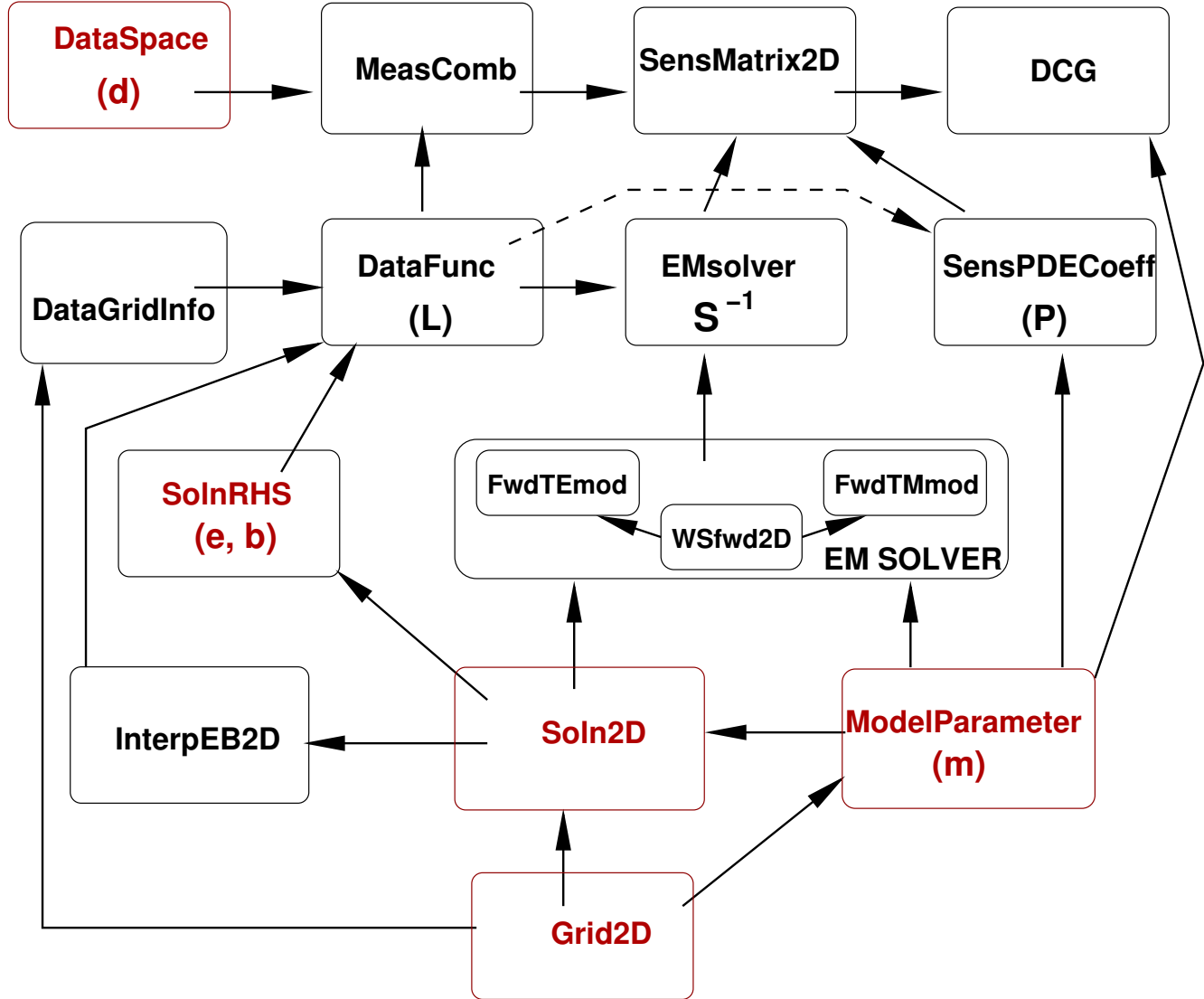


Figure 1: Dependence structure for 2D modular inversion system. Upper level routines are designed to be completely general (and in particular will work for both 2D and 3D problems). Second level is specific to the 2D MT problem, but is mostly independent of the specific modeling approach; much of this code could also be reused for other 2D problems, e.g., active source. The model parameter module is self-contained, with all attributes of model parameter objects private.

2.2 Development of a modular inversion and modeling system

The second part of this project has been to develop a modular inversion system for 2D and 3D frequency domain EM inverse problems. The basis for this modular approach is the observation that a wide range of different inversion algorithms can be expressed using a common abstract notation, in terms of a relatively small number of symbols representing objects such as parameter and data vectors (\mathbf{m}, \mathbf{d}), the non-linear mapping between model and data ($\mathbf{f}(\mathbf{m})$), model or data covariances ($\mathbf{C}_d, \mathbf{C}_m$), and the sensitivity matrix or Jacobian (\mathbf{J} with ij element $J_{ij} = \partial f_i / \partial m_j$). The sensitivity matrix in turn can be formally represented in terms of objects such as EM field vectors (\mathbf{e}), differential operators ($\mathbf{S}_{\omega, \mathbf{m}}$), data functionals (\mathbf{f}) and their linearizations (\mathbf{L}, \mathbf{Q}), and a sparse operator (\mathbf{P}), which depends on details of the model parameterization and grid discretization. A fairly general explicit expression for the Jacobian is

$$\mathbf{J} = \mathbf{L} \mathbf{S}_{\omega, \mathbf{m}}^{-1} \mathbf{P} + \mathbf{Q}. \quad (1)$$

A more detailed definition of all of the component operators, and examples for specific inverse problem implementations are given in the attached preprint.

Some of the objects represented by these abstract symbols are very large dense matrices (e.g., $\mathbf{S}_{\omega, \mathbf{m}}^{-1}$) which could not plausibly be formed; and many of the inversion algorithms discussed do not require actually forming the Jacobian \mathbf{J} . In fact, these, and even the sparse matrices $\mathbf{P}, \mathbf{L}, \mathbf{Q}$ should be viewed as operators which can act on an arbitrary input vector, producing an output of the appropriate type (e.g., multiplication by $\mathbf{S}_{\omega, \mathbf{m}}^{-1}$ means “apply the EM forward solver to the input source/boundary condition vector, to get an electric field object”). A modular system requires development of classes of model parameter, electric field solution, and data space objects, together with the necessary operators. Individual modules have been constructed to be as self-contained as possible, and are coded so as to be as independent as possible from other modules. For example, data space objects connect to model parameters so indirectly that these objects never need know any details about the internal representation or methods of the other.

We have developed a prototype of all of the essential modules for both 2D and 3D MT inversion, using an object oriented approach in Fortran 95. The dependency structure for the 2D version is given in Figure 1. The top level of the modular system consists of data space, data sensitivity, and ultimately a number of inversion routines (only one, DCG for “Dataspace Conjugate Gradients” is illustrated in the figure). Basic object classes that are manipulated at this top level, such as $\mathbf{m}, \mathbf{e}, \mathbf{d}$ are treated as abstract data types (ADTs), with generic names, and a set of basic methods for creation, destruction, I/O, basic linear algebra and dot products. Different instances of these objects and methods are implemented for different EM problems (in particular, for 2D and for 3D) in the lower level modules. Derived data types in Fortran 95 are used to define the basic data objects, symbolic operators like “+”, “-”, “*”, “/” or “.dot.” are overloaded so that top level code for different inversion approaches can be written almost as abstractly and succinctly as in a text book discussion of geophysical inverse methods.

Just below the top-level generic inversion modules sits a layer of modules that are specific to a specific class of EM inverse problem, e.g., 3D MT vs 2D MT vs. active source 2D vs. global observatory. The modules we have created (so far for 2D and 3D MT) provide templates for other EM inverse problems. Significant changes from the finite difference modeling

approach we have focused on would probably require some changes in modules at this level, but even with significant changes (e.g., to a finite element or integral equations modeling approach) much of the code at this level could be reused. Adding new measurement operators would require additions at this level. One key module at this level defines implementation of the generic EM solution space data type used as an ADT by the top level routines. Note that for 2D the basic EM solution could be a scalar field (e.g., E_x or H_x for TE or TM modes), together with tags to keep track of mode, and pointers to transmitter and model parameters. For 3D the basic EM solution would have to include 3 component vector fields for 2 source polarizations (corresponding to different boundary conditions). To allow an abstract treatment of EM solutions at the top level both vector fields must be merged into a single EM solution object; e.g., evaluation of linearized (and non-linear) data functionals requires both solutions. Similarly it is necessary to define objects to represent data functionals with sufficient abstraction to accommodate multi-source vector fields, single component scalar fields, and other possibilities. Generic methods for these data types (including basic, linear algebra, and dot products) are also required to support the top level routines. Other key modules at this second, problem specific, level include: data functional definitions; EM solver control, providing an abstract interface for initializing and using the equation solver; managing and storing EM solution objects needed for further computations; and implementation of model parameter/solver coefficient sensitivity mappings.

Finally, we note that the model parameterization is constructed as a single module, allowing the model parameter to be treated as an ADT by all other modules. Components of the model parameterization model include the basic definition of the model parameter data type, basic operations (including model space linear algebra and dot products), plus model covariances, and operations for mapping the conductivity to the coefficients of the EM differential equations. Ultimately constraints on model parameters could be allowed for by modifying this module, although we have not implemented any examples of this yet. By keeping all attributes of model parameters private we guarantee that the model parameterization can be modified independently of all other modules.

2.3 An efficient data space OCCAM scheme : 2D MT example

Siripunvaraporn et al. (2002) have demonstrated that with a data space approach a Gauss-Newton (GN) type inversion scheme (Data Space Occam, or DASSOC) based on a full sensitivity calculation is at least feasible, even in 3D. However, the full sensitivity calculation is still very time consuming, and the need to store the full Jacobian rather severely limits model discretization, and dataset size. Conjugate gradient (CG) methods would appear still to have substantial advantages for 3D problems, as they avoid direct computation and storage of the data sensitivities, and also avoid the need to directly store or invert the large model (or data) space matrix.

We have developed a new efficient algorithm for inversion of multi-frequency EM data that combines elements of both the GN and CG approaches. To generate one search direction CG and related schemes require two solutions of the governing differential equations per frequency. These calculations effectively generate data sensitivities for one linear data contrast for each frequency. By saving these sensitivities a good approximation to the full Jacobian can be built up in a comparatively small number of CG steps, allowing use of approximate GN methods.

Using this idea we have develop a new algorithm combining elements of the standard CG and “Occam” minimum structure inversions. Tests on synthetic data for a two-dimensional magnetotelluric problem show the method provides essentially the same solutions as those obtained using a GN method requiring full calculation of the Jacobian, but at a fraction of the computational cost. The new approach is also more efficient than the standard CG approach, and can be readily generalized to multi-frequency three-dimensional EM inverse problems. The approach and the initial tests are described in further detail in the attached preprint.

3 Publications resulting from this grant

- Siripunvaraporn, W., G. Egbert, and Y. Lenbury, Numerical accuracy of magnetotelluric modeling: A comparison of finite difference approximations, *Earth, Planets, Space*, 54, 721-726, 2002.
- Siripunvaraporn, W., Uyeshima, M., and Egbert, G., Three-Dimensional Inversion for Network-Magnetotelluric Data, *Earth Planets Space* 56, 893-902, 2004.
- Siripunvaraporn, W., Egbert, G., Lenbury, Y., and Uyeshima, M., Three-Dimensional Magnetotelluric Inversion: Data Space Method, *Phys. Earth Planet. Inter.* 150, 3-14, 2005a.
- Siripunvaraporn, W, G, Egbert, and M. Uyeshima, Interpretation of 2-D Magnetotelluric Profile Data with 3-D Inversion: Synthetic Examples, *Geophys. J. Int.* 160, 804-814, 2005b.

A fifth publication, presently in review for publication in *Inverse Problems* is

- Egbert, G.D., and W. Siripunvaraporn, Efficient Inversion of Multi-Frequency Electromagnetic Data: A Hybrid Conjugate Gradients/Gauss-Newton Algorithm, 2006.

A preprint of this manuscript is included at the end of this report.

Efficient Inversion of Multi-Frequency Electromagnetic Data: A Hybrid Conjugate Gradients/Gauss-Newton Algorithm

G D Egbert¹, W Siripunvaraporn²

¹ College of Oceanic and Atmospheric Sciences, Oregon State University, Corvallis, OR USA

² Department of Physics, Mahidol University, Bangkok, Thailand

E-mail: egbert@coas.oregonstate.edu

Abstract. We present a new more efficient algorithm for regularized inversion of multi-frequency electromagnetic (EM) data. To generate one search direction conjugate gradient (CG) methods used for penalty functional minimization require two solutions of the governing differential equations per frequency. These calculations effectively generate data sensitivities for one linear data contrast for each frequency. By saving these sensitivities a good approximation to the full Jacobian of the penalty functional can be built up in a comparatively small number of CG steps, allowing use of approximate Gauss-Newton (GN) methods. Using this idea we develop a new algorithm combining elements of the standard CG and “Occam” minimum structure inversions. Tests on synthetic data for a two-dimensional magnetotelluric problem show the method provides essentially the same solutions as those obtained using a GN method requiring full calculation of the Jacobian, but at a fraction of the computational cost. The new approach is also more efficient than the standard CG approach, and generalizes readily to multi-frequency three-dimensional EM inverse problems.

AMS classification scheme numbers: 86A22,86A25,86-08,35R30

Submitted to: *Inverse Problems*

1. Introduction

One of the most widely used approaches in electromagnetic (EM) geophysics for inversion of field data is to seek an Earth conductivity model that fits the observations adequately, while minimizing some measure of model size or roughness. This is accomplished by defining a penalty functional including data misfit and model norm terms, and then minimizing this using some mathematical optimization method. Although general purpose derivative-free methods such as simulated annealing and genetic algorithms have been tried (e.g., Schultz et al., 1993; Everett and Schultz, 1993), these methods have not yet proven practical for realistic two- and three-dimensional EM inverse problems, where the number of unknown model parameters is very large. For real datasets there are presently two practical options: imaging schemes based on approximation of the forward problem (e.g., Torres-Verdin et al., 2000; Xie et al., 2000; Zhdanov et al., 2000; Zhdanov and Tartaras, 2002), and gradient-based optimization methods based on more rigorous linearization of the model parameter-data relationship. Our focus here is on the later "linearized inversion" approach.

The most direct approach to linearized inversion is to calculate the full Jacobian of the data mapping, and then use some variant on a Gauss-Newton (GN) approach; e.g., Parker (1994) discusses several widely used algorithms in the general context of geophysical inverse problems. This direct approach has long been used for one-dimensional EM (and other geophysical) inverse problems (e.g., Constable et al. (1987) and Smith and Booker (1988) discuss EM applications). These sensitivity based methods have also been used successfully for routine two-dimensional EM inversion (e.g., deGroot-Hedlin and Constable, 1990; Uchida, 1993; Siripunvaraporn and Egbert, 2000), and with some modifications, for three-dimensional problems (Sasaki, 2004; Siripunvaraporn et al., 2004, 2005). However, for multi-dimensional EM inverse problems calculation of the full Jacobian requires numerical solution of many forward problems, and direct application of a GN-type approach also typically requires solution of a very large dense system of linear equations. Thus, other approaches such as conjugate gradients (CG; e.g., Mackie and Madden, 1993; Newman and Alumbaugh, 1996; Rodi and Mackie, 2001), non-linear CG (NLCG; Rodi and Mackie, 2001) or quasi-Newton methods (QN; Newman and Boggs, 2004; Avdeev, 2005), which do not require calculation or storage of the full Jacobian, have become the preferred approach for multi-dimensional inversion of EM data.

CG, NLCG, and QN (generically CG) are limited memory optimization schemes which require only the gradient of the penalty functional to define a new search direction. In fact, these gradients are linear combinations of the data sensitivities required for the Jacobian calculation, so both GN and CG approaches require similar sorts of numerical calculations. The key difference is that GN uses the full Jacobian to approximate the Hessian of the penalty functional, while CG type approaches build up second derivative information about penalty functional curvature sequentially over a series of search steps (e.g., Press et al., 1986).

Methods such as CG avoid storage and direct inversion of large dense matrices and thus unquestionably have a major advantage over GN with respect to computer memory requirements. However, CG approaches converge slowly relative to GN and thus still require long run times and many solutions to the forward EM problem (e.g., Newman and Boggs, 2004). In fact, in the common situation where EM data are collected for multiple frequencies or sources, it is far from clear that standard CG schemes actually require fewer forward modeling steps than a GN approach. Consider the case where there are M model parameters with a total of $N_d = N_s N_f$ data collected at N_s sites and N_f frequencies. As discussed more fully below, computation of the full $M \times N_d$ Jacobian nominally requires solution of N_d modeling problems. In the CG approach (Mackie and Madden, 1993; Newman and Alumbaugh, 1997) the $M \times M$ system of GN equations is solved iteratively in an increasing sequence of subspaces constructed from successively calculated directions of steepest descent. Each additional direction added to the search space requires solving the forward modeling equations twice for each frequency. Essentially the same computations would generate $2N_f$ rows of the sensitivity matrix, so searching $N_s/2$ directions in the model space will require roughly the same amount of CPU time as computing the Jacobian. In fact, as shown in Siripunvaraporn and Egbert (2000), only a portion of the sensitivity matrix is required to form a very good approximation to the Hessian of the penalty functional. Thus to be efficient with respect to operation counts this CG scheme would have to achieve convergence in a small number of iterations, significantly less than half the total number of sites. Convergence tests for 2D magnetotelluric (MT) inverse problems reported in Rodi and Mackie (2001) and Siripunvaraporn and Egbert (2006) suggest that such rapid convergence will not generally be achieved.

Our thesis in this paper is that information about the Jacobian is actually obtained for an N_f dimensional subspace of the effective model space in each search step. Standard CG approaches collapse this to a single model space search direction, effectively discarding much useful information about penalty functional curvature. Our goals here are to further elucidate this statement, and then to begin exploring alternative linearized inversion schemes which make better use of the multi-frequency data sensitivity information that is generated in each search step of a CG scheme. We begin in section 2 with a brief review of the linearized inversion methods typically applied to EM data, including both GN and CG approaches. Then, in section 3 we review the CG algorithm in more detail, and show that these schemes can be slightly modified to generate a sequence of data sensitivities for linear combinations of data. At the cost of storing the results of these calculations separately for each frequency, a very good approximation to the Hessian of the penalty functional can be built in a comparatively small number of search steps. In section 4 we develop a new inversion scheme based on these ideas. In the new approach we save sensitivities for linear combinations of data generated by applying CG separately to each frequency, and then use these in a reduced data space minimum structure inversion. In section 5 we present synthetic data tests for a 2D MT problem, demonstrating that the new approach produces results essentially

identical to a 2D minimum structure inversion based on a full Jacobian calculation, but at a fraction of the computational cost. The new scheme, which generalizes readily to 3D problems, is also more efficient than previously proposed CG approaches. Memory requirements for the new approach, while significantly greater than for CG, are still modest enough to allow reasonable sized 3D inverse problems to be practical on a modern desktop workstation with several gigabytes of memory.

2. Linearized EM inversion

Aspects of a number of previously described linearized EM inversion methods enter into the development of the new algorithm described here. We thus begin with a brief review of key elements of these now standard approaches using a consistent notation. Siripunvaraporn and Egbert (2000), Rodi and Mackie (2001), Newman and Boggs (2004), and Avdeev (2005) provide further details on these and related methods.

We consider regularized inversion based on gradient-based minimization of a penalty functional of the form

$$\mathcal{J}(\mathbf{m}, \mathbf{d}) = (\mathbf{d} - \mathbf{f}(\mathbf{m}))^T \mathbf{C}_d^{-1} (\mathbf{d} - \mathbf{f}(\mathbf{m})) + \nu (\mathbf{m} - \mathbf{m}_0)^T \mathbf{C}_m^{-1} (\mathbf{m} - \mathbf{m}_0) \quad (1)$$

to recover, in a stable manner, Earth conductivity distributions which provide an adequate fit to a data vector \mathbf{d} of dimension N_d . In (1) \mathbf{C}_d is the covariance of data errors, \mathbf{m} is the M -dimensional model parameter vector (e.g., $\ln(\sigma)$ for discrete blocks), $\mathbf{f}(\mathbf{m})$ defines the forward mapping (which must be computed numerically), \mathbf{m}_0 is a prior or first guess model parameter, ν is a regularization parameter, and \mathbf{C}_m (or more properly $\nu^{-1}\mathbf{C}_m$) defines the model covariance or regularization term. In practice \mathbf{C}_d is always taken to be diagonal, so by a simple rescaling of the data and forward mapping ($\mathbf{C}_d^{-1/2}\mathbf{d}$, $\mathbf{C}_d^{-1/2}\mathbf{f}$), we may eliminate \mathbf{C}_d^{-1} from the definition of \mathcal{J} . Similarly we can transform \mathbf{m} (and the forward mapping \mathbf{f}) to eliminate the prior model, so without loss of generality we may assume $\mathbf{m}_0 = 0$, $\mathbf{C}_d = \mathbf{I}$.

Before proceeding we note that many of the computations in frequency domain EM problems are most efficiently implemented (and described) using complex arithmetic, but the model conductivity parameter \mathbf{m} is real. Data might be complex (an impedance) or real (an apparent resistivity and/or phase). For simplicity we will assume that all data are real, i.e., real and imaginary parts of a complex impedance element will be treated as separate elements of the real data vector \mathbf{d} , and that all operators map real vectors to real vectors. These mappings can always be recast in this form, although in practice many of these will actually be implemented most efficiently using complex arithmetic. See the Appendix for further details.

The derivative of \mathbf{f} with respect to the model parameters is the $N_d \times M$ sensitivity matrix (or Jacobian) \mathbf{J} , with ij element $J_{ij} = \partial f_i / \partial m_j$. Search for a minimizer of (1) using \mathbf{J} is iterative, as for example in the classical GN method. Denoting by \mathbf{m}_n the model parameter at the n th iteration and \mathbf{J}^n the sensitivity matrix evaluated at \mathbf{m}_n the

penalty functional for $\mathbf{m}_n + \delta\mathbf{m}$ can be approximated as

$$\mathcal{J}_n = (\mathbf{d}' - \mathbf{J}^n \delta\mathbf{m})^T (\mathbf{d}' - \mathbf{J}^n \delta\mathbf{m}) + \nu (\mathbf{m}_n + \delta\mathbf{m})^T \mathbf{C}_m^{-1} (\mathbf{m}_n + \delta\mathbf{m}), \quad (2)$$

where $\mathbf{d}' = \mathbf{d} - \mathbf{f}(\mathbf{m}_n)$. As pointed out in Haber et al. (2000) and Rodi and Mackie (2001), the full quadratic Taylor series approximation of \mathcal{J} at \mathbf{m}_n includes an additional term involving second derivatives of the data with respect to model parameters. This calculation is actually feasible (e.g., Haber et al., 2000), but the value of this complication is debatable (e.g., Press et al., 1986). Differentiating \mathcal{J}_n with respect to $\delta\mathbf{m}$ and setting the result to zero leads to the usual $M \times M$ system of equations for a stationary point

$$(\mathbf{J}^{nT} \mathbf{J}^n + \nu \mathbf{C}_m^{-1}) \delta\mathbf{m} = \mathbf{J}^{nT} \mathbf{d}' - \nu \mathbf{C}_m^{-1} \mathbf{m}_n, \quad (3)$$

which can be solved for $\delta\mathbf{m}$ to yield a new trial solution $\mathbf{m}_{n+1} = \mathbf{m}_n + \delta\mathbf{m}$. As discussed in Parker (1994) this basic linearized scheme generally requires some form of step length damping for stability (e.g., a Levenberg-Marquardt approach; Marquardt, 1963).

The Occam approach (Constable et al., 1987) is based on a slight modification of (3), which, with $\hat{\mathbf{d}} = \mathbf{d} - \mathbf{f}(\mathbf{m}_n) + \mathbf{J}^n \mathbf{m}_n$, can be re-written

$$(\mathbf{J}^{nT} \mathbf{J}^n + \nu \mathbf{C}_m^{-1}) \mathbf{m} = \mathbf{J}^{nT} \hat{\mathbf{d}}. \quad (4)$$

Although \mathbf{m}_{n+1} is obtained directly as the solution to (4) the result is exactly equivalent to solving (3) for the change in the model at step $n + 1$ and adding the result to \mathbf{m}_n . However, in the Occam scheme step length control is achieved by varying ν , computing a series of trial solutions \mathbf{m}_ν to (4), and solving the forward problem for each \mathbf{m}_ν to evaluate the actual data misfit. In phase I of the Occam process ν is chosen to minimize data misfit; as the scheme converges ν is chosen to minimize the model norm while keeping the misfit constant (phase II). See Constable et al. (1987) and Parker (1994) for further details. A significant advantage of the Occam approach is that the regularization parameter ν is determined as part of the search process, and at convergence one is assured that the solution attains at least a local minimum of the model norm $\|\mathbf{m}\| = (\mathbf{m}^T \mathbf{C}_m^{-1} \mathbf{m})^{1/2}$, subject to the data fit attained (Parker, 1994). In other approaches the regularization parameter ν must be varied independently to choose an optimal value (e.g., Newman and Alumbaugh 2000; Newman and Boggs, 2004).

There is a slightly different way to implement the Occam scheme in the data space (Siripunvaraporn and Egbert, 2000). The solution to (4) can be written as

$$\mathbf{m}_{n+1} = \mathbf{C}_m \mathbf{J}^{nT} \mathbf{x} \quad (5)$$

where the coefficients \mathbf{x} satisfy

$$(\mathbf{J}^n \mathbf{C}_m \mathbf{J}^{nT} + \nu^{-1} \mathbf{I}) \mathbf{x} = \hat{\mathbf{d}}. \quad (6)$$

The equivalence of the data space approach is readily verified by substituting (5) and (6) into (4) and simplifying. The N_d columns of $\mathbf{C}_m \mathbf{J}^{nT}$, which are just the data sensitivities smoothed by the model covariance, are referred to as the *representers* of the linearized data functionals (e.g., Parker, 1994). Eq. (5) means that at step n the Occam solution

is a linear combination of the representers of the data functionals, linearized about the solution for the previous iteration. The coefficients in the representer expansion are calculated by solving the $N_d \times N_d$ system of equations in the data space (6) so we refer to this as the “data space Occam” or DASOCC method (Siripunvaraporn and Egbert, 2000; Siripunvaraporn et al., 2005). Direct solution of (3) or (4) involves an $M \times M$ system of equations in the model space. If the model is heavily over-parameterized (as will generally be the case for 3D inversion) N_d may be much smaller than M , making the DASOCC scheme significantly more efficient than equivalent model space approaches. Perhaps more importantly, the data space perspective can provide key insights into possible efficiencies in methods for optimizing the penalty functional, as we shall show.

For all of the methods described above we have assumed implicitly that the full Jacobian \mathbf{J} has already been computed—a task which requires substantial computational effort for multi-dimensional EM problems, since the equivalent of one forward solution is required for each row (or column) of \mathbf{J} . Methods for solving (3) with CG avoid explicitly calculating and storing \mathbf{J} , or the cross-product matrices which occur in (3), (4) or (6). This approach, which has been implemented for EM problems by Mackie and Madden (1993), Zhang et al. (1995), Newman and Alumbaugh (1996), Rodi and Mackie (2001), and others, requires computation of the matrix vector product $\mathbf{A}\mathbf{m} = [\mathbf{J}^{nT}\mathbf{J}^n + \nu\mathbf{C}_m^{-1}]\mathbf{m}$ for an arbitrary model space vector \mathbf{m} . As we show in the appendix, the full sensitivity matrix for general EM problems can be formally written in the form

$$\mathbf{J}^n = \mathbf{L}\mathbf{S}_{\mathbf{m}_n}^{-1}\mathbf{P} + \mathbf{Q}, \quad (7)$$

where \mathbf{L} represents the measurement process (i.e., the linearized data functionals); $\mathbf{S}_{\mathbf{m}_n}^{-1}$ denotes the EM solution operator for conductivity distribution \mathbf{m}_n ; and \mathbf{P} and \mathbf{Q} are, respectively, $N_e \times M$ and $N_d \times M$ matrices related to the model parameterization (of dimension M), numerical EM solution representation (of dimension N_e), and the data functionals (of dimension N_d). The matrix vector product $\mathbf{J}^n\mathbf{m}$ can thus be written

$$\mathbf{J}^n\mathbf{m} = \mathbf{L}\mathbf{S}_{\mathbf{m}_n}^{-1}\mathbf{P}\mathbf{m} + \mathbf{Q}\mathbf{m}. \quad (8)$$

The matrices \mathbf{P} and \mathbf{Q} are in general sparse, and formation of the products $\mathbf{P}\mathbf{m}$ and $\mathbf{Q}\mathbf{m}$ does not entail significant computation. The only computationally demanding task required to form the product of an arbitrary model space vector \mathbf{m} with the Jacobian \mathbf{J}^n is thus solution of one EM forward problem. Similarly, the transpose of the Jacobian can be written

$$\mathbf{J}^{nT} = \mathbf{P}^T[\mathbf{S}_{\mathbf{m}_n}^T]^{-1}\mathbf{L}^T + \mathbf{Q}^T, \quad (9)$$

so multiplication of an arbitrary data space vector by \mathbf{J}^{nT} requires one solution of the transposed EM system, which is in fact essentially equivalent to solution of the forward problem (see the Appendix). Thus, as Mackie and Madden (1993) show, multiplication of an arbitrary model space vector by $\mathbf{J}^{nT}\mathbf{J}^n + \nu\mathbf{C}_m^{-1}$ can be accomplished without forming or storing \mathbf{J}^n , at the cost of two solutions of the EM problem.

Expressions (7) and (9) clearly show that multiplication of a data or model vector by \mathbf{J}^{nT} or \mathbf{J}^n , respectively, can be accomplished with essentially the same calculations required to compute a single data sensitivity; indeed, as will be critical to our development below, the product $\mathbf{J}^{nT}\mathbf{c}$, is just the sensitivity for the linear data contrast (i.e., the linear combination of data vector components) $\mathbf{c}^T\mathbf{d}$. Formation of the full sensitivity matrix, using either (7) or (9) requires solving the EM equations once for each frequency for each of the M columns of \mathbf{P} or once for each of the N_d columns of \mathbf{L}^T , respectively.

The CG scheme we have just described iteratively solves for $\delta\mathbf{m}$ or \mathbf{m}_{n+1} based on linearization in the vicinity of \mathbf{m}_n . This iterative solver is then embedded in a further outer loop (over n) to solve the non-linear inverse problem. A variant on this CG approach, is to apply CG directly as a non-linear quadratic optimization scheme (e.g., Press et al., 1986) to minimize (2) (e.g., Rodi and Mackie, 2001; Newman and Boggs, 2004; Avdeev, 2005). With this NLCG approach one must evaluate the gradient of (1) with respect to variations in model parameters \mathbf{m} :

$$\left. \frac{\partial \mathcal{J}}{\partial \mathbf{m}} \right|_{\mathbf{m}_n} = -\mathbf{J}^{nT}(\mathbf{d} - \mathbf{f}(\mathbf{m}^n)) + \nu \mathbf{C}_m^{-1} \mathbf{m}_n. \quad (10)$$

NLCG utilizes essentially the same basic computational steps as required for solving the linearized equations (3) (i.e., the forward problem must be solved to evaluate $\mathbf{f}(\mathbf{m})$, and the adjoint problem must be solved to multiply the residual $\mathbf{d} - \mathbf{f}(\mathbf{m})$ by \mathbf{J}^{nT}), and additional forward modeling steps are required for line minimization along each CG search direction (e.g., Newman and Alumbaugh, 2000). Quasi-Newton schemes (QN; e.g., Nocedal and Wright, 1999; Newman and Boggs, 2004; Haber, 2005; Avdeev, 2005) provide an alternative approach to NLCG for direct minimization of (1), with similar advantages with regard to storage and computation of the Jacobian, and similar computational requirements. NLCG essentially eliminates the “inner loop” required by CG, and has thus become (along with QN) more commonly used for practical 3D EM inversion. However, GN exhibits quadratic convergence, and typically only a very small number of outer loop steps (3-8) are required for convergence. Many more steps are required for convergence of NLCG, so the relative efficiency of these two approaches is hard to assess a priori. In the only study directly comparing these two approaches on an EM problem that we are aware of, Rodi and Mackie (2001) found that for 2D MT inversion CG and NLCG performed similarly, in terms of total run time required for convergence. The new approach proposed here builds on the CG and DASOCC schemes; however the basic ideas discussed below could most probably also be adapted to improve efficiency of NLCG or QN.

So far in our review of inversion algorithms we have not explicitly addressed the issue that is central to the methods developed in this paper: in most cases EM data are obtained for a wide range of frequencies, or for a number of different transmitter configurations. Here we focus on the case of multiple frequencies, as for example in the case of MT data. Allowing for multiple transmitters presents similar issues and opportunities, though there are also enough differences to warrant a separate discussion.

Assuming that there are data for N_f frequencies $\omega_1, \dots, \omega_{N_f}$, each application of a matrix-vector operation such as (8) (or its transpose) actually requires solving N_f forward problems, one for each frequency.

To be more explicit, partition the data vector and sensitivity matrix into separate blocks for each frequency

$$\mathbf{d} = (\mathbf{d}_1^T \dots \mathbf{d}_{N_f}^T) \quad \mathbf{J} = (\mathbf{J}_1^T \dots \mathbf{J}_{N_f}^T), \quad (11)$$

and partition \mathbf{L} and \mathbf{Q} accordingly. Then the full sensitivity matrix can be written explicitly as

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_1 \\ \vdots \\ \mathbf{J}_{N_f} \end{pmatrix} = \begin{pmatrix} \mathbf{L}_1 \mathbf{S}_{\omega_1, \mathbf{m}_n}^{-1} \mathbf{P}_1 + \mathbf{Q}_1 \\ \vdots \\ \mathbf{L}_{N_f} \mathbf{S}_{\omega_{N_f}, \mathbf{m}_n}^{-1} \mathbf{P}_{N_f} + \mathbf{Q}_{N_f} \end{pmatrix}. \quad (12)$$

Note that the matrices \mathbf{P}_l and \mathbf{Q}_l generally depend on the solution for frequency ω_l , and thus are also different for each frequency. A key observation is that to multiply a full data space vector (e.g., the residual $\mathbf{r} = \mathbf{d} - \mathbf{f}(\mathbf{m})$ in the gradient equation (10)) one forms the sum

$$\mathbf{J}^T \mathbf{r} = \sum_{l=1}^{N_f} \mathbf{J}_l^T \mathbf{r}_l, \quad (13)$$

where \mathbf{r}_l is the part of the data space vector \mathbf{r} corresponding to frequency l . As noted above, $\mathbf{J}^T \mathbf{r}$ is the derivative (sensitivity) of the linear combination of data $\mathbf{r}^T \mathbf{d}$ with respect to the model parameters \mathbf{m} . Moreover, $\mathbf{J}_l^T \mathbf{r}_l$ is the sensitivity of the linear combination $\mathbf{r}_l^T \mathbf{d}_l$ of data restricted to frequency l . Thus, each step in a CG or NLCG scheme actually requires solution of the forward and adjoint problems N_f times, and results in computation of N_f linear combinations of data sensitivities, one for each frequency. Our objective here is to develop more efficient ways to use the component sensitivities for each frequency ($\mathbf{J}_l^T \mathbf{r}_l$) in the inversion processes.

Before proceeding, some brief comments on the regularization term in (1) are appropriate. We have formulated the penalty using \mathbf{C}_m , a smoothing operator which is usually interpreted as the covariance of the model parameters, providing *a priori* constraints on the magnitude and spatial smoothness of conductivity variations. In most model space formulations the regularization term is formulated instead in terms of a roughening operator, e.g., writing $\mathbf{C}_m^{-1} = \mathbf{D}^T \mathbf{D}$, where \mathbf{D} is a discrete representation of a gradient or higher order derivative operator. This approach seems, at first blush, to be simpler and more natural for the model space formulation, since, as \mathbf{D} is sparse, multiplication by \mathbf{C}_m^{-1} in (3) or (4) is then trivial to implement and very efficient. In contrast, with the data space approach \mathbf{C}_m^{-1} never appears in any equations (except for the definition of the penalty functional) or in any computations; rather only multiplication by \mathbf{C}_m is required (see (5), (6)). Hence, for the data space approach it is more natural to formulate the regularization in terms of \mathbf{C}_m . Although this matrix is generally not sparse and may not be practical to invert, it is easy to construct computationally efficient positive definite symmetric smoothing operators that

reasonably characterize prior information about model smoothness (e.g., Egbert et al., 1994; Siripunvaraporn and Egbert, 2000; Chua and Bennett, 2001).

In fact, it is possible, and in some important respects advantageous, to use the same approach in the model space. In terms of the transformed model parameter $\tilde{\mathbf{m}} = \mathbf{C}_{\mathbf{m}}^{-1/2}\mathbf{m}$, (4) can be written

$$(\mathbf{C}_{\mathbf{m}}^{1/2}\mathbf{J}^{nT}\mathbf{J}^n\mathbf{C}_{\mathbf{m}}^{1/2} + \nu\mathbf{I})\tilde{\mathbf{m}} = \mathbf{C}_{\mathbf{m}}^{1/2}\mathbf{J}^{nT}\hat{\mathbf{d}}. \quad (14)$$

After solving (14) for $\tilde{\mathbf{m}}$ we can easily recover $\mathbf{m} = \mathbf{C}_{\mathbf{m}}^{1/2}\tilde{\mathbf{m}}$. All of these computations require only multiplication by the smoothing operator $\mathbf{C}_{\mathbf{m}}^{1/2}$ (i.e, half of the smoothing of $\mathbf{C}_{\mathbf{m}}$). Thus one can arrange the calculations so that $\mathbf{C}_{\mathbf{m}}^{-1}$ is also never required with the model space formulation, and the fact that $\mathbf{C}_{\mathbf{m}}$ may not be readily invertible does not pose any difficulty. There is good reason to believe that this modified formulation actually leads to a much better conditioned system, as now the term $\nu\mathbf{I}$ is truly stabilizing in the inversion. As pointed out in Haber and Ascher (2001), $\mathbf{C}_{\mathbf{m}}^{-1}$ generally has an unbounded spectrum, making $\mathbf{J}^{nT}\mathbf{J}^n + \nu\mathbf{C}_{\mathbf{m}}^{-1}$ very poorly conditioned. Indeed, standard practice with a model space formulation is to precondition (3) using an approximation to the operator $\mathbf{C}_{\mathbf{m}} = [\mathbf{D}^T\mathbf{D}]^{-1}$ (e.g., approximate solution of Poisson's equation; Rodi and Mackie, 2001; Haber and Ascher, 2001). Formulation of the regularization in terms of a model covariance $\mathbf{C}_{\mathbf{m}}$, and then transforming to (14) achieves the same improvement in conditioning at essentially no computational cost.

Note that in terms of the transformed model parameters the Jacobian is

$$\frac{\partial \mathbf{f}}{\partial \tilde{\mathbf{m}}} = \mathbf{J}\mathbf{C}_{\mathbf{m}}^{1/2} = \tilde{\mathbf{J}}, \quad (15)$$

and the model and data space Gauss-Newton equations are reduced to

$$(\tilde{\mathbf{J}}^{nT}\tilde{\mathbf{J}}^n + \nu\mathbf{I})\tilde{\mathbf{m}} = \tilde{\mathbf{J}}^{nT}\mathbf{d} \quad (\tilde{\mathbf{J}}^n\tilde{\mathbf{J}}^{nT} + \nu^{-1}\mathbf{I})\mathbf{x} = \mathbf{d}. \quad (16)$$

To simplify notation in the following development we use these transformed canonical forms, omitting the tildes, as well as the superscript outer loop iteration index n .

3. Conjugate Gradients

We now review the conjugate gradients (CG) algorithm used for iterative solution of the positive definite symmetric GN equations in the data space. Everything presented here is quite standard (e.g., Press et al., 1986; Golub and Van Loan, 1989), but perhaps not completely familiar to many who use CG as a tool either for minimization, or for iterative solution of linear equations. These internal details of the CG method will be essential to our subsequent development.

3.1. Solving Symmetric Systems with CG

To solve the symmetric positive definite system of equations $\mathbf{Ax} = \mathbf{b}$, CG generates a sequence of approximate solutions to the linear system of equations by iterative

minimization of the quadratic form

$$\mathcal{Q}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{b}^T \mathbf{x} \quad (17)$$

along a sequence of conjugate search directions \mathbf{h}_k . The search starts from an arbitrary first guess \mathbf{x}_0 , and with $\mathbf{h}_0 = 0$. The gradient of \mathcal{Q} at \mathbf{x}_k is

$$\mathbf{g}_{k+1} = \mathbf{A} \mathbf{x}_k - \mathbf{b}, \quad (18)$$

and the next (conjugate) search direction is formed as the linear combination of this direction of steepest descent and the previous search direction

$$\mathbf{h}_{k+1} = \mathbf{g}_{k+1} - \gamma_k \mathbf{h}_k, \quad \gamma_k = \frac{\mathbf{g}_{k+1}^T \mathbf{A} \mathbf{h}_k}{\mathbf{h}_k^T \mathbf{A} \mathbf{h}_k}. \quad (19)$$

The minimum of $\mathcal{Q}(\mathbf{x})$ along \mathbf{h}_{k+1} (starting from \mathbf{x}_k) is

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda_{k+1} \mathbf{h}_{k+1}, \quad \lambda_{k+1} = \frac{\mathbf{h}_{k+1}^T \mathbf{g}_{k+1}}{\mathbf{h}_{k+1}^T \mathbf{A} \mathbf{h}_{k+1}}. \quad (20)$$

It is readily shown that the search directions \mathbf{h}_k generated by this process are conjugate in the sense that

$$\mathbf{h}_k^T \mathbf{A} \mathbf{h}_{k'} = 0 \quad k \neq k', \quad (21)$$

and that the gradient vectors are all mutually orthogonal

$$\mathbf{g}_k^T \mathbf{g}_{k'} = 0 \quad k \neq k'. \quad (22)$$

By combining (18) and (20) we obtain

$$\mathbf{g}_{k+1} = \mathbf{g}_k - \lambda_k \mathbf{A} \mathbf{h}_k, \quad (23)$$

so taking one step in the CG solver scheme only requires multiplication of the previous search direction \mathbf{h}_k by the matrix \mathbf{A} , plus forming some inner products and linear combinations of vectors.

Some basic properties of the sequence of CG solutions \mathbf{x}_k will be essential to subsequent developments. First, each \mathbf{x}_k is a linear combination of the search directions and the initial trial solution \mathbf{x}_0 , so in the common implementation with $\mathbf{x}_0 = 0$

$$\begin{aligned} \mathbf{x}_k \in \mathcal{H}_k = \text{Sp}\{\mathbf{h}_1, \dots, \mathbf{h}_k\} = \\ \text{Sp}\{\mathbf{b}, \mathbf{A} \mathbf{b}, \mathbf{A}^2 \mathbf{b}, \dots, \mathbf{A}^{k-1} \mathbf{b}\}. \end{aligned} \quad (24)$$

Furthermore, it is easily seen that the gradient vectors also span the space \mathcal{H}_k . Finally, from the two conjugacy conditions and the fact that \mathcal{Q} is minimized over each of the search directions it is readily verified that \mathbf{x}_k minimizes $\mathcal{Q}(\mathbf{x})$ over the subspace \mathcal{H}_k . The CG scheme thus can be viewed as an efficient sequential scheme for generating a sequence of (Krylov) subspaces $\mathcal{H}_k \subset \mathcal{H}_{k+1}$, and at each stage minimizing \mathcal{Q} over the subspace.

3.2. CG Solution of the Data Space Equations

We now consider more explicitly using CG to solve the data space GN equations (with $\mathbf{C}_m = \mathbf{I}$)

$$(\mathbf{J}\mathbf{J}^T + \nu^{-1}\mathbf{I})\mathbf{x} = \hat{\mathbf{d}}, \quad (25)$$

for the coefficients in the representer expansion $\mathbf{m} = \sum_k \mathbf{x}_k \mathbf{j}_k$. Application of the data space conjugate gradient (DCG) scheme to EM inversion is described in Siripunvaraporn and Egbert (2006); applications to ocean data assimilation are given in Egbert et al. (1994), and Chua and Bennett (2001). With $\mathbf{C}_m = \mathbf{I}$ the representer \mathbf{j}_i is just the data sensitivity for observation i (i.e., the i^{th} column of \mathbf{J}^T). In the notation of the general CG development above $\mathbf{A} = \mathbf{J}\mathbf{J}^T + \nu^{-1}\mathbf{I}$ and $\hat{\mathbf{d}} = \mathbf{b}$, and we have for the Krylov subspace after a total of K steps

$$\mathcal{H}_K = Sp\{\hat{\mathbf{d}}, \mathbf{J}\mathbf{J}^T\hat{\mathbf{d}}, \dots, (\mathbf{J}\mathbf{J}^T)^{K-1}\hat{\mathbf{d}}\} \quad (26)$$

Let $(\mathbf{u}_1, \dots, \mathbf{u}_K)$ be an orthonormal basis for \mathcal{H}_K and let \mathbf{U}_K be the orthogonal $N_d \times K$ matrix with these columns. The step K CG solution can be written $\mathbf{x}_K = \mathbf{U}_K \mathbf{c}$ for some K -dimensional vector \mathbf{c} , and since \mathbf{x}_K minimizes \mathcal{Q} over \mathcal{H}_K , \mathbf{c} must minimize

$$\begin{aligned} \tilde{\mathcal{Q}}_K(\mathbf{c}) &= \frac{1}{2}(\mathbf{U}_K \mathbf{c})^T (\mathbf{J}\mathbf{J}^T + \nu^{-1}\mathbf{I}) \mathbf{U}_K \mathbf{c} - \hat{\mathbf{d}}^T \mathbf{U}_K \mathbf{c} = \\ &= \frac{1}{2} \mathbf{c}^T (\tilde{\mathbf{J}}_K \tilde{\mathbf{J}}_K^T + \nu^{-1}\mathbf{I}) \mathbf{c} - \tilde{\mathbf{d}}_K^T \mathbf{c}, \end{aligned} \quad (27)$$

where

$$\tilde{\mathbf{J}}_K = \mathbf{U}_K^T \mathbf{J} \quad \tilde{\mathbf{d}}_K = \mathbf{U}_K^T \hat{\mathbf{d}}. \quad (28)$$

Now $\tilde{\mathbf{d}}_K$ is just a set of K linear combinations of the components of the vector $\hat{\mathbf{d}}$, and the corresponding columns of $\tilde{\mathbf{J}}_K^T = \mathbf{J}^T \mathbf{U}_K$ are the sensitivities for these K data contrasts. Furthermore, the step K CG model parameter obtained in the data space CG scheme is just

$$\mathbf{m} = \mathbf{J}^T \mathbf{x}_K = \mathbf{J}^T \mathbf{U}_K \mathbf{c} = \tilde{\mathbf{J}}_K^T \mathbf{c} = \tilde{\mathbf{J}}_K (\tilde{\mathbf{J}}_K \tilde{\mathbf{J}}_K^T + \nu^{-1}\mathbf{I})^{-1} \tilde{\mathbf{d}}_K^T. \quad (29)$$

Thus, the step K data space CG solution can be viewed as a reduced data space solution where the K linear combinations of the original data $\tilde{\mathbf{d}}_K$, and the corresponding reduced set of data sensitivities are used in place of the full set of data and sensitivities in (5) and (6). The CG scheme works with relatively modest values of K due to the significant redundancy in the data, with low signal-to-noise ratios in most linear data contrasts. Data redundancy also forms the basis for the truncated singular value decomposition (SVD) approach commonly used in geophysical inverse problems (e.g., Parker, 1994). With this approach the SVD of the linear model parameter data mapping is approximated using K leading modes of the SVD of the Jacobian as

$$\mathbf{d} = \mathbf{J}\mathbf{m} + \epsilon = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \mathbf{m} + \epsilon \approx \mathbf{U}_K \mathbf{\Lambda}_K \mathbf{V}_K^T + \epsilon. \quad (30)$$

As is readily verified, the truncated SVD approach is equivalent to projecting the data into a K -dimensional subspace $\tilde{\mathbf{d}}_K = \mathbf{U}_K^T \mathbf{d}$, and then minimizing the standard penalty

functional using this reduced data vector in place of \mathbf{d} . In fact, the search directions generated by data space CG will form a good approximation to the leading eigenvectors of $\mathbf{J}^T \mathbf{J}$, which are identical to the left singular vectors \mathbf{U} of \mathbf{J} . See the Appendix for further discussion of the reduced data space approach, and its relationship to the reduced basis approach of Siripunvaraporn and Egbert (2000; see also Parker and Shure 1982; Parker et al., 1987; Oldenburg *et al.*; 1993, Egbert and Erofeeva, 2002; Kurapov et al., 2003).

The data space CG scheme can of course be easily modified to actually generate $\tilde{\mathbf{d}}_K$ and $\tilde{\mathbf{J}}_K$. At each outer loop step the product $\mathbf{J}\mathbf{J}^T \mathbf{r}_k$ must be formed, where $\mathbf{r}_0 = \hat{\mathbf{d}}$ and $\mathbf{r}_{k+1} = \mathbf{J}\mathbf{J}^T \mathbf{r}_k$. As noted above, $\mathbf{J}^T \mathbf{r}_k$ is just the sensitivity for the linear data contrast $\mathbf{r}_k^T \mathbf{d}$. Saving these vectors for K steps, the $N_d \times K$ matrix $\mathbf{R}_K = (\mathbf{r}_1 \dots \mathbf{r}_K)$ can be formed and orthonormalized, e.g.,

$$\mathbf{U}_K = \mathbf{R}_K (\mathbf{R}_K^T \mathbf{R}_K)^{-\frac{1}{2}}. \quad (31)$$

Then we can also form

$$(\mathbf{J}^T \mathbf{r}_1 \dots \mathbf{J}^T \mathbf{r}_K) (\mathbf{R}_K^T \mathbf{R}_K)^{-\frac{1}{2}} = \mathbf{J}^T \mathbf{R}_K (\mathbf{R}_K^T \mathbf{R}_K)^{-\frac{1}{2}} = \mathbf{J}^T \mathbf{U}_K = \tilde{\mathbf{J}}_K^T, \quad (32)$$

and $\tilde{\mathbf{d}}_K = \mathbf{U}_K^T \mathbf{d}$. Of course, the data space CG scheme already computes \mathbf{m} satisfying (29) with a sequential scheme that does not require saving \mathbf{r}_k and $\mathbf{J}^T \mathbf{r}_k$, or doing any of the computations of (31) and (32). However, with these additional steps (29) could be solved for a range of damping parameters ν —i.e., an Occam scheme, where ν is used for step length control, could be used without redoing all of the computations required for K data sensitivity calculations per frequency for each ν .

For a multi-frequency problem the sensitivities $\mathbf{J}^T \mathbf{r}_k, k = 1, \dots, K$ are each the sum of N_f data sensitivities $\mathbf{J}_l^T \mathbf{r}_{kl}$ for linear combinations of data at frequencies $\omega_l, l = 1, \dots, N_f$ (i.e., $\mathbf{r}_{kl}^T \mathbf{d}_l$), each obtained by solving the adjoint equation for the appropriate frequency. Thus, instead of saving the sums $\mathbf{J}^T \mathbf{r}_k$ we could save the individual terms $\mathbf{s}_{kl} = \mathbf{J}_l^T \mathbf{r}_{kl}$, along with the corresponding single transmitter data space vectors \mathbf{r}_{kl} . Then we would have sensitivities \mathbf{s}_{kl} for a much larger data subspace, of dimension KN_f instead of K . The reduced data space approximate solutions analogous to (29) might then be expected to provide an improved approximation to the full data space GN system (25) with the same or fewer forward and adjoint solutions and a small amount of additional computation. Indeed, if we define

$$\tilde{\mathcal{H}}_K = Sp\{\mathbf{r}_{kl}, l = 1, \dots, N_f, k = 1, \dots, K\}, \quad (33)$$

then clearly $\mathcal{H}_K \subset \tilde{\mathcal{H}}_K$ so this modified approach would be guaranteed to yield a lower value of the quadratic form \mathcal{Q} . It does not necessarily follow that the approximate solution to (6) will be improved (e.g., in terms of reducing relative solution error), but this argument certainly does suggest that there are substantial opportunities for improved efficiency in the CG approach, allowing use of a much larger data space and an improved approximation to the Jacobian with relatively minor additional computation, together with a more significant increase in memory usage.

4. An Hybrid CG-Occam Inversion Scheme

In the previous section we showed how the data space CG scheme could be used to generate a sequence of data contrasts \mathbf{r}_{kl} , and corresponding sensitivities $\mathbf{J}_l^T \mathbf{r}_{kl}$, which could be used in an Occam-style regularized inversion. There are other, plausibly better, ways to generate a similar sequence. We consider one such approach in the following; a number of variants and refinements are readily imagined.

Our algorithm is based on the DASOCC scheme which is summarized in Figure 1a. However, instead of using the full dataset and computing the full Jacobian, we apply K steps in a minimization scheme analogous to model space CG to generate data sensitivities for a total of KN_f linear data contrasts. These are then used in a reduced data space Occam scheme, solving a system analogous to (29).

More explicitly, in the first stage we consider solution of the model space GN system for a single frequency with $\nu = 0$

$$\mathbf{J}_l^T \mathbf{J}_l \mathbf{m} = \mathbf{J}_l^T \hat{\mathbf{d}}_l. \quad (34)$$

This is equivalent to minimizing the frequency l data misfit, based on the linearized data mapping \mathbf{J}_l ,

$$\mathcal{Q}_l(\mathbf{m}) = (\mathbf{J}_l \mathbf{m} - \hat{\mathbf{d}}_l)^T (\mathbf{J}_l \mathbf{m} - \hat{\mathbf{d}}_l). \quad (35)$$

Of course there is no unique solution to (34) (assuming the number of data for a single frequency is less than the number of model parameters), but we can still apply CG, which will converge to the model parameter \mathbf{m} satisfying (34) with smallest L_2 norm. The idea here is that by taking the first K steps to minimizing (35) CG will generate linear combinations of data with the (approximately) greatest sensitivity to variations in the model parameters (in the neighborhood of the solution from the previous iteration).

At the first step in the CG scheme for frequency l we have

$$\mathbf{h}_{1l} = -\mathbf{J}_l^T \hat{\mathbf{d}}_l, \quad (36)$$

and at step k we must compute

$$\mathbf{A} \mathbf{h}_{k-1,l} = \mathbf{J}_l^T \mathbf{J}_l \mathbf{h}_{k-1,l}. \quad (37)$$

Setting $\mathbf{r}_{k+1,l} = \mathbf{J}_l \mathbf{h}_{k,l}$, $\mathbf{r}_{1l} = -\hat{\mathbf{d}}_l$, we see that CG minimization of (35) generates a sequence of data space vectors $\{\mathbf{r}_{kl}, k = 1, K\}$, and corresponding data sensitivities $\mathbf{s}_{kl} = \mathbf{J}_l^T \mathbf{r}_{kl}, k = 1, K$ for the linear data contrasts $\mathbf{r}_{kl}^T \hat{\mathbf{d}}_l, k = 1, K$. Orthogonalizing as above (but now separately for each frequency)

$$\begin{aligned} \mathbf{U}_{Kl} &= \mathbf{R}_{Kl} (\mathbf{R}_{Kl}^T \mathbf{R}_{Kl})^{-\frac{1}{2}} \\ \tilde{\mathbf{J}}_{Kl}^T &= (\mathbf{J}_l^T \mathbf{r}_{1l} \dots \mathbf{J}_l^T \mathbf{r}_{Kl}) (\mathbf{R}_{Kl}^T \mathbf{R}_{Kl})^{-\frac{1}{2}} = (\mathbf{s}_{1l} \dots \mathbf{s}_{Kl}), \end{aligned} \quad (38)$$

and setting

$$\tilde{\mathbf{J}}_K = (\tilde{\mathbf{J}}_{K1}^T \dots \tilde{\mathbf{J}}_{KN_f}^T) \quad \mathbf{U}_K = (\mathbf{U}_{K1}^T \dots \mathbf{U}_{KN_f}^T), \quad (39)$$

we proceed with the reduced data space Occam scheme. That is, using the linear combinations of data $\tilde{\mathbf{d}}_k = \mathbf{U}_K \mathbf{d}$, and the corresponding sensitivities (columns of $\tilde{\mathbf{J}}_K$),

a reduced data space equation (29) is solved for trial values of ν , and \mathbf{m}_ν is chosen to minimize (over ν) the RMS misfit to the full dataset. Then $\mathbf{m}_{n+1}\mathbf{m}_\nu$ is used as the background solution for the next GN iteration. As with the standard Occam scheme, once the desired normalized RMS is achieved, phase II of the Occam process is begun, with the regularization parameter ν now chosen to minimize the model norm subject to achieving the desired misfit (Parker, 1994). Pseudo-code for this hybrid scheme is given in Figure 1b.

5. Synthetic Data Examples

To test the scheme described in the previous section we consider the transverse electric (TE) mode 2D MT inversion. The 2D MT TE problem is formulated in Cartesian coordinates, with electrical conductivity of the Earth varying in the y and z directions, and with electric fields oriented in the x direction. In this case the quasi-static approximation to Maxwell's equations with a time dependence of $e^{-i\omega t}$ can be reduced to a 2-D scalar diffusion equation for the along strike electric field

$$\nabla^2 E_x - i\omega\mu_0\sigma E_x = 0, \quad (40)$$

subject to the usual MT boundary conditions (uniform sources far above the Earth's surface, E_x going to zero at depth). Given E_x the orthogonal component of the magnetic field can be computed from Faraday's law as $H_y = (i\omega\mu)^{-1}\partial_z E_x$. The fundamental data for the TE mode inverse problem are impedances $Z_{xy} = E_x/H_y$, measured at N_s sites and N_f frequencies $\omega_1, \dots, \omega_{N_f}$.

We generated synthetic TE mode impedances, for the two conductivity models of Figure 2 using a finite difference modeling code, with grids of dimension 100×40 , including 10 air layers above the Earth's surface. For the inversion each Earth cell in the finite difference grid was treated as an independent parameter (so $M = 3000$). Complex impedances were generated for $N_s = 30$ sites at $N_f = 10$ frequencies, evenly spaced logarithmically between 0.001 - 1.0 hz. There are thus 60 real data for each frequency, and the full data vector \mathbf{d} is of length $N_d = 600$. Random noise with a magnitude of 5% was added before inversion using both the DASOCC scheme of Siripunvaraporn and Egbert (2000), and the hybrid scheme of the previous section. For the hybrid scheme we set $K = 5$, so each outer loop step required solving each of the forward and adjoint problems a total of $KN_f = 50$ times, the equivalent of 100 sensitivity calculations. Because sensitivities for real and imaginary parts of the impedance can be calculated together (with a single adjoint solution; see the Appendix), computation of the full Jacobian needed for DASOCC required three times as many calls to the EM solver (300 total). Storage of the sensitivities is reduced by a factor of 6 with the hybrid approach.

Final inverse model results using the two approaches for the simpler test case are compared in Figure 3. Essentially the same final model is obtained in both cases, and the same number of outer loop iterations is required. The convergence of the two algorithms is illustrated in Figure 4, where we plot misfit as a function of ν for each outer

loop iteration in the Occam process. Convergence properties of the two approaches are essentially identical.

Figures 5 and 6 give comparisons of inverse models, and Occam misfit curves for the more complex model of Figure 2b. For this particular data set neither DASOCC nor the hybrid scheme quite fits the data adequately. The full sensitivity calculation DASOCC scheme does achieve a slightly smaller misfit, and the hybrid scheme required two additional iterations to get close to the misfit achieved by DASOCC. However, it would be difficult to argue that the final inverse solution is superior, even if the outer loop iterations for the hybrid approach were stopped sooner. To compare to a CG type approach, for the two cases we take 4 and 6 outer loop iterations, requiring computations the equivalent of 5 search steps per iteration. In a direct comparison of DCG and DASOCC, Siripunvaraporn and Egbert (2006) showed that for a problem of comparable size and complexity, convergence of CG to even an approximate solution to the data space equations required for each outer loop iteration required 15-20 steps. They further show that DCG generally requires at least as many forward solutions as DASOCC, so the hybrid scheme should also be significantly faster than DCG. NLCG may be more efficient than CG (but see Rodi and Mackie, 2001), but unless a very effective preconditioner can be found, it seems unlikely that NLCG would achieve convergence on this problem in the $4 - 6 \times 5 = 20 - 30$ steps required by the hybrid CG-DASOCC scheme.

6. Discussion

We have shown that CG schemes for solving EM inverse problems generate data sensitivities for a sequence of linear data contrasts. For multi-frequency problems such as MT, sensitivities can be saved for each frequency separately, and the saved sensitivities provide a good approximation to the leading singular vectors of the Jacobian. Conventional CG approaches ignore multi-frequency aspects of the problem, discarding useful information by collapsing multi-frequency sensitivities to a single model space search direction. We have considered one possible use of this multi-frequency information, applying a GN approach (Occam) to minimize the penalty functional in a reduced data space defined by the CG search directions. The proposed scheme converges in a small number of outer loop iterations, because it is in essence a Newton type method, and thus makes use of substantial second derivative information in minimizing the penalty functional. A further advantage of the new algorithm is that as with Occam (Constable et al., 1987), the regularization parameter ν is computed as part of the inversion process, so that smoothness (in the sense of the model norm \mathbf{C}_m) is minimized for the achieved data misfit. For CG or NLCG adjusting the regularization parameter, or finding a true norm minimizing solution, would require multiple inversions.

One issue that deserves some discussion is that we explicitly fit only a reduced data vector. To what extent does this make our approach sub-optimal? And what is the advantage of this, compared to a simpler approach such as fitting only a subset of

the available data? It is instructive to compare this reduced data space scheme to the reduced basis approach for large linear inverse problems described by Parker and Shure (1982), Parker et al. (1987), Oldenburg *et al.*, (1993), and Parker (1994), and applied to non-linear MT inversion by Siripunvaraporn and Egbert (2000). For this discussion we consider the simplest canonical minimum norm linear inverse problem, with data and model parameters related by $\mathbf{d} = \mathbf{J}\mathbf{m}$, and the inverse solution obtained by minimizing the penalty functional

$$(\mathbf{d} - \mathbf{J}\mathbf{m})^T(\mathbf{d} - \mathbf{J}\mathbf{m}) + \mathbf{m}^T\mathbf{m}. \quad (41)$$

Both reduced basis and reduced data space approaches can be described in terms of an $N_d \times K$ orthogonal matrix \mathbf{U} . In the reduced data space approach used here we consider the K -dimensional reduced data vector $\mathbf{U}^T\mathbf{d}$, simply replacing \mathbf{J} by $\tilde{\mathbf{J}} = \mathbf{U}^T\mathbf{J}$. The optimal solution has the usual form

$$\hat{\mathbf{m}}_{rd} = \tilde{\mathbf{J}}^T(\tilde{\mathbf{J}}\tilde{\mathbf{J}}^T + \mathbf{I})^{-1}\tilde{\mathbf{d}}, \quad (42)$$

and is a linear combination of the columns of $\tilde{\mathbf{J}}$.

For the reduced basis approach we instead apply the orthogonal matrix to \mathbf{J} to obtain a reduced model space basis, the columns of $\tilde{\mathbf{J}}^T = \mathbf{J}^T\mathbf{U}$. The original penalty functional (41) is then minimized over the model subspace spanned by this reduced set of basis vectors. The reduced basis solution is readily derived as

$$\hat{\mathbf{m}}_{rb} = \tilde{\mathbf{J}}^T[\tilde{\mathbf{J}}\mathbf{J}^T\tilde{\mathbf{J}}\tilde{\mathbf{J}}^T + \tilde{\mathbf{J}}\tilde{\mathbf{J}}^T]^{-1}\tilde{\mathbf{J}}\mathbf{J}^T\mathbf{d}. \quad (43)$$

Thus the reduced basis solution also only fits data in a K -dimensional subspace of the data space, i.e. the data projected onto the column span of the $N_d \times K$ matrix $\tilde{\mathbf{J}}\tilde{\mathbf{J}}^T = \mathbf{J}\mathbf{J}^T\mathbf{U}$.

In the usual application of the reduced basis approach representers (i.e., columns of \mathbf{J}) for a simple subset of the data are used, so (after possibly reordering data) $\mathbf{U}^T = (\mathbf{I} \mid \mathbf{0})$. It is clear that in this case a reduced basis approach would be very different from the reduced data space approach using the same \mathbf{U} . In the former case all data vector components would be fit, while the latter would amount to discarding all but K of the data before inversion. Obviously, the reduced basis approach would be much preferable in this case. In the Appendix we show that the reduced basis and reduced data space inverse solutions are identical if and only if the columns of the orthogonal matrix \mathbf{U} are linear combinations of a set of K eigenvectors of the Gram matrix $\Gamma = \mathbf{J}\mathbf{J}^T$. The matrix \mathbf{U} consisting of the largest K eigenvectors of Γ provides the optimal reduction of the data space ($\mathbf{U}^T\mathbf{d}$), and also the optimal reduction of the model space basis vectors ($\mathbf{J}^T\mathbf{U}$). We use CG to generate a reduced data space, defined by the span of the first K search directions. This space will include very good approximations to the leading eigenvectors of Γ ; this is the basis of Lanczos methods for approximate solution of eigenvalue problems (e.g., Golub and Van Loan, 1989). The reduced data spaces generated through CG should thus approach the optimal reductions to a K dimensional subspace that can be achieved through a truncated eigenvector approach,

and also mean that we approximate a reduced basis approach, using an optimal reduction in the model space basis.

A number of extensions and modifications of the basic method presented here could lead to further improvements in efficiency. For example, for 2D MT inversion of both TE and TM modes, linear combinations of data sensitivities could be generated and saved separately for the two modes. This idea could obviously be generalized to other problems with multiple source geometries. Also, in our initial illustrative test we used a constant value of K . By allowing this to vary with frequency (e.g., computing fewer sensitivities for lower frequencies) further efficiency might be realized. In addition, as we show in the Appendix, for complex data such as impedances each step in the CG scheme used to generate the data contrasts, and corresponding sensitivities, actually generates two sensitivities corresponding roughly to $\mathbf{J}^T \mathbf{r}_k$, and $\mathbf{J}^T \tilde{\mathbf{r}}_k$, where $\tilde{\mathbf{r}}_k$ corresponds to the real vector representation of $i\mathbf{r}_k$. In contrast to the data vectors \mathbf{r}_k generated by the CG process, the complementary $\tilde{\mathbf{r}}_k$ vectors are not likely to be particularly important directions in the data space, so it is not clear that these additional sensitivities are actually worth storing. However, this issue deserves more careful analysis and testing.

Finally, we note that the scheme presented here is almost entirely implemented in the data space, so there is essentially no dependency on special features of the 2D MT problem. The scheme generalizes to 3D EM problems with multi-frequency data (in particular 3D MT) with essentially no changes. For 3D MT the factor of 5-10 reduction in run times and memory that can be achieved, relative to a full GN inversion scheme, are much more significant. At present 3D EM inversion is at the edge of practicality. Without access to very substantial computing resources significant compromises in model size and resolution, dataset size, or both are required. An order of magnitude improvement in computational efficiency for these sorts of EM inverse problems would obviously go a long way toward making routine 3D inversion practical for a much larger group of users. The computational efficiency that likely can be realized by making better use of all of the data sensitivity information that is generated in a CG search in a multi-frequency EM inversion appears to have the potential to realize such an improvement in efficiency.

Acknowledgments

GDE was supported by DOE grants DE-FG0202-ER15318 and DE-FG02-06ER15819. WS was supported by the Thailand Research Fund (RSA4780021).

Appendix A.

Appendix A.1. Computation of Data Sensitivities for EM Problems

Here we consider the sensitivity calculation for general EM problems, omitting reference to the iteration index n . The numerical discretization of the frequency domain EM

differential equation is written generically as

$$\mathbf{S}_{\omega, \mathbf{m}} \mathbf{e} = \mathbf{b} \quad (\text{A.1})$$

where \mathbf{b} gives appropriate boundary and forcing terms for the particular EM problem, and \mathbf{e} is the N_e component vector representing the discretized electric and/or magnetic fields (or perhaps potential functions). $\mathbf{S}_{\omega, \mathbf{m}}$ is an $N_e \times N_e$ matrix which depends on the frequency ω and an M dimensional model parameter \mathbf{m} . Given a solution \mathbf{e} to (A.1) data can always be written in the general form

$$d_j = f_j(\mathbf{m}) + \epsilon_j = \psi_j(\mathbf{e}(\mathbf{m}), \mathbf{m}) + \epsilon_j \quad (\text{A.2})$$

where ψ_j is some generally non-linear, but usually simple, function of the components of \mathbf{e} (and possibly \mathbf{m}), and ϵ_j represents data error.

With this general setup we have, by the chain rule

$$J_{jk} = \frac{\partial f_j}{\partial m_k} = \frac{\partial \psi_j}{\partial m_k} = \sum_l \frac{\partial \psi_j}{\partial e_l} \frac{\partial e_l}{\partial m_k} + \frac{\partial \psi_j}{\partial m_k} \quad (\text{A.3})$$

Let $\mathbf{F} = \partial \mathbf{e} / \partial \mathbf{m}$ be the $N_e \times M$ matrix with elements $F_{lk} = \partial e_l / \partial m_k$, and \mathbf{L}, \mathbf{Q} be the partial derivative matrices

$$L_{jl} = \left. \frac{\partial \psi_j}{\partial e_l} \right|_{\mathbf{e}, \mathbf{m}} \quad Q_{jk} = \left. \frac{\partial \psi_j}{\partial m_k} \right|_{\mathbf{e}, \mathbf{m}} \quad (\text{A.4})$$

Then the Jacobian can be written in matrix notation as

$$\mathbf{J} = \mathbf{L}\mathbf{F} + \mathbf{Q} \quad (\text{A.5})$$

Note that \mathbf{F} , \mathbf{L} and \mathbf{Q} in general depend on the model parameters \mathbf{m} . The rows of \mathbf{L} and \mathbf{Q} (one for each observation) are generally very sparse, supported only on a few nodes surrounding the corresponding data site. When the observation functionals are independent of the model parameters (as they often are) $\mathbf{Q} \equiv \mathbf{0}$. The j th row of \mathbf{L} represents the linearized data functional, which is applied to the perturbation in the EM solution to compute the perturbation in d_j . Although coding \mathbf{L} and \mathbf{Q} can be quite involved for realistic EM data functionals, calculation of \mathbf{F} presents the only computational challenge.

To derive a formal general expression for \mathbf{F} take the derivative of (A.1) with respect to the model parameters \mathbf{m} . Assuming \mathbf{b} remains fixed (if this is not true minor modifications are necessary; Newman and Boggs (2004) treat this additional complication), and allowing that as \mathbf{m} is varied the solution \mathbf{e} also varies, we obtain

$$\mathbf{S}_{\omega, \mathbf{m}} \left[\left. \frac{\partial \mathbf{e}}{\partial \mathbf{m}} \right|_{\mathbf{m}=\mathbf{m}_0} \right] = - \frac{\partial \mathbf{S}_{\omega, \mathbf{m}} \mathbf{e}_0}{\partial \mathbf{m}}, \quad (\text{A.6})$$

or

$$\mathbf{S}_{\omega, \mathbf{m}_0} \mathbf{F} = \mathbf{P}. \quad (\text{A.7})$$

In (A.6) \mathbf{e}_0 is the solution to (A.1) for model parameter \mathbf{m}_0 . The $N_e \times M$ matrix \mathbf{P} depends on details of both the numerical model implementation and the conductivity parameterization, but is in general inexpensive to calculate.

As an explicit example, consider the quasi-static frequency domain 3D electric field induction equation

$$\nabla \times \nabla \times \mathbf{E} - i\omega\mu\sigma\mathbf{E} = \mathbf{s} \quad (+ \text{ BC}), \quad (\text{A.8})$$

approximated with finite differences on a staggered grid (e.g., Yee, 1966; Smith, 1996; Siripunvaraporn et al., 2002). Model conductivity is parameterized in the simplest way, with an independent parameter for each computational grid cell. With the staggered grid approach electric field components are defined on grid cell edges, and conductivity appears only in the second (diagonal) term in (A.8), where it multiplies the electric field components. This requires defining conductivity on cell edges; a physically consistent (current conserving) way to do this is to use the volume weighted average of the conductivity of the four adjacent cells to define the edge conductivities needed for the staggered grid discretization of (A.8). With this choice of mapping from model parameters to discrete operator coefficients, it is readily verified that

$$\mathbf{P} = \mathbf{diag}(i\omega\mu\mathbf{e}_0)\mathbf{W} \quad (\text{A.9})$$

where \mathbf{W} is the $N_e \times M$ linear map which averages cell conductivities onto grid edges. Note that if instead of choosing cell conductivities as the model parameter we chose instead log conductivity, this non-linear parameter mapping has to be linearized, and we find $\mathbf{P} = \mathbf{diag}(i\omega\mu\mathbf{e}_0)\mathbf{W}\mathbf{diag}(\mathbf{m}_0)$. It is straightforward to work out \mathbf{P} for other cases, such as for the EM equations formulated in terms of magnetic fields, or for modified parameterizations of the conductivity. In all cases the computational burden involved in multiplying by \mathbf{P} is negligible.

Putting together (A.5) and (A.7) we have a formal expression for the $N_d \times M$ Jacobian

$$\mathbf{J} = \mathbf{L}\mathbf{S}_{\omega,\mathbf{m}}^{-1}\mathbf{P} + \mathbf{Q}. \quad (\text{A.10})$$

To compute all of \mathbf{J} would thus appear to require solving the induction equation once for each of the M columns of \mathbf{P} . However, simply taking the transpose of (A.10) we obtain

$$\mathbf{J}^T = \mathbf{P}^T[\mathbf{S}_{\omega,\mathbf{m}}^T]^{-1}\mathbf{L}^T + \mathbf{Q}^T, \quad (\text{A.11})$$

so the sensitivity matrix can in fact be obtained by solving the transposed discrete EM system N_d times, the usual "reciprocity" trick for efficient calculation of sensitivities (e.g., Rodi, 1976, de Lugao *et al.*, 1997). Note that the EM equations are self-adjoint (except for time reversal) with respect to the usual L^2 inner product (i.e., reciprocity holds), so on a uniform grid $\mathbf{S}_{\omega,\mathbf{m}}$ is symmetric (though there are some technical issues regarding boundary conditions). For more general grids the fact that the EM operator is self-adjoint implies $\mathbf{S}_{\omega,\mathbf{m}}^T = \mathbf{V}\mathbf{S}_{\omega,\mathbf{m}}\mathbf{V}^{-1}$, where \mathbf{V} is a diagonal matrix of integration volume elements for the natural discrete representation of the L_2 integral inner product. (Note that other, equivalent, ways to symmetrize the discretized system are often used in practice.) Thus, solution of the transposed (or adjoint) system requires only minor modifications to the forward solver.

Appendix A.2. Reduced Data Space and Reduced Basis Inversion

Here we show that reduced data space and reduced basis inversions defined by the $N_d \times K$ orthonormal matrix \mathbf{U} are identical if and only if the columns of \mathbf{U} span the same subspace as a set of K eigenvectors of $\Gamma = \mathbf{J}\mathbf{J}^T$ (or equivalently, the left singular vectors of \mathbf{J} ; another equivalent condition is that the columns of \mathbf{U} span an invariant subspace of Γ). Indeed, if the columns of \mathbf{U} and \mathbf{W}_K have the same span, where columns of \mathbf{W}_K are eigenvectors of Γ then $\mathbf{U} = \mathbf{W}_K \mathbf{Q}$ where \mathbf{Q} is $K \times K$ orthonormal. Substituting this expression for \mathbf{U} and the SVD of $\mathbf{J} = \mathbf{W}\mathbf{\Lambda}\mathbf{V}^T$ into (42) and (43) and simplifying, it is readily verified that $\hat{\mathbf{m}}_{rd} = \hat{\mathbf{m}}_{rb}$. The converse is also readily demonstrated. From (42) and (43) we see for the two inverse solutions to be identical for all data vectors \mathbf{d} , \mathbf{U} and $\Gamma\mathbf{U}$ must span the same data subspace. Then there is a $K \times K$ matrix \mathbf{A} for which $\mathbf{U}\mathbf{A} = \Gamma\mathbf{U}$, implying $\mathbf{A} = \mathbf{U}^T\Gamma\mathbf{U}$ is symmetric and positive definite, and can thus be decomposed as $\mathbf{A} = \mathbf{Q}\Theta\mathbf{Q}^T$ where Θ is diagonal and \mathbf{Q} is $K \times K$ orthonormal. But then we have $[\mathbf{U}\mathbf{Q}]\Theta = \Gamma[\mathbf{U}\mathbf{Q}]$, i.e., the columns of $[\mathbf{U}\mathbf{Q}]$ are eigenvectors of Γ .

Appendix A.3. Complex Data, Real Model Parameters

The frequency domain EM differential equations are formulated in the complex domain, and the EM solutions are complex fields. For simplicity data are also generally complex, e.g., impedance components. On the other hand the unknown model parameter, electrical conductivity, is generally taken to be real. As a result care is required in interpreting the sensitivity matrix \mathbf{J} . Literally $\mathbf{S}_{\omega,\mathbf{m}}$ represents a matrix with complex entries, so the formal expression for the Jacobian in (A.10) also represents a complex matrix which we here call $\bar{\mathbf{J}}$. The real (imaginary) part of the product $\bar{\mathbf{J}}\delta\mathbf{m}$ gives the sensitivity of the real (imaginary) part of the data, so one can separate the real and imaginary parts of $\bar{\mathbf{J}}$ and the data vector to obtain a real matrix \mathbf{J} (with twice as many rows) which maps real model parameters to real data. This is the approach we have implicitly taken in our discussion of inversion methods, in which we have assumed that all matrices were real.

Using reciprocity to calculate $\bar{\mathbf{J}}$ with (A.11) requires solution of the adjoint equations once for each data vector element. The result of the formal calculation is an M -dimensional complex vector, one row of $\bar{\mathbf{J}}$. Thus sensitivities for real and imaginary parts of the data are calculated with a single solution of the transposed complex system $\mathbf{S}_{\omega,\mathbf{m}}^T$. Hence, if we take N_d to be the total number of real data, which actually consist of real and imaginary parts of complex data, the number of model solutions required for complete calculation of the Jacobian is $N_d/2$. Note that apparent resistivity and phase data can also be treated essentially as real and imaginary parts of the natural log of the complex impedance, so sensitivities for both of these responses can also be computed with a single call to the solver.

Next consider computation of something like $\mathbf{J}^T\mathbf{r}$. The actual code based on (A.11) implements multiplication of complex vectors $\bar{\mathbf{r}}$ by $\bar{\mathbf{J}}$ resulting in a complex

M -dimensional vector. To compute the desired real matrix vector product $\Re[\bar{\mathbf{J}}^T]\mathbf{r}_r + \Im[\bar{\mathbf{J}}^T]\mathbf{r}_i$, where $\mathbf{r}_r, \mathbf{r}_i$ correspond to real and imaginary data components, we must set $\bar{\mathbf{r}} = \mathbf{r}_r - i\mathbf{r}_i$, and take the real part of the complex product $\Re[\bar{\mathbf{J}}\bar{\mathbf{r}}]$. Key to our development of the new inversion algorithm is our observation that $\mathbf{J}^T\mathbf{r}$ is the sensitivity for a particular linear data contrast, $\mathbf{r}^T\mathbf{d}$. In the process of computing this we obtain also $\Im[\bar{\mathbf{J}}\bar{\mathbf{r}}]$. In fact, this is the sensitivity for the data contrast $\tilde{\mathbf{r}}^T\mathbf{d}$, where $\tilde{\mathbf{r}}$ is the real data vector corresponding to $i\bar{\mathbf{r}}$. Thus we actually generate sensitivities for a two dimensional subspace of the real data space when we compute $\bar{\mathbf{J}}^T\bar{\mathbf{r}}$. The additional sensitivity could in principal be used to improve the approximation of the Jacobian, at the cost of saving $\Im[\bar{\mathbf{J}}\bar{\mathbf{r}}]$.

References

- Avdey, D. B. 2005 Three-dimensional electromagnetic modelling and inversion from theory to application *Surv. Geophys.* **26**,767-799
- Chua, B.S., and A.F. Bennett 2001 An inverse ocean modeling system *Ocean Modeling* **3** 137-165
- Constable, C. S., Parker, R. L., and Constable, C.G. 1987 Occam's inversion: A practical algorithm for generating smooth models from electromagnetic sounding data *Geophysics* **52** 289-300
- de Lugao, P., Portniaguine, O., and Zhdanov, M.S. 1997 Fast and stable two-dimensional inversion of magnetotelluric data *Jour. Geomag. Geoelec.* **49** 1437-1454
- deGroot-Hedlin, C., and Constable, S. 1990 Occam's inversion to generate smooth, two-dimensional models from magnetotelluric data *Geophysics* **55** 1613-1624
- Egbert, G. D., and S. Yu. Erofeeva 2002 Efficient Inverse Modeling of Barotropic Ocean Tides *J. Atmos. Ocean. Tech.* **19** 183-2004
- Egbert, G. D., Bennett, A. F., Foreman, M. G. G. 1994 Topex/Poseidon tides estimated using a global inverse model *J. Geophys. Res.* **99** 24821-24852
- Everett, M. E., and A. Schultz 1993 Two dimensional nonlinear magnetotelluric inversion using a genetic algorithm *Jour. Geomag. Geoelec.* **45** 1013-1026
- Golub, C. F., Van Loan, C. F. 1989 *Matrix Computations, 2nd ed.* (Baltimore: Johns Hopkins University Press)
- Kurapov, A. L., G. D. Egbert, J. S. Allen, R. N. Miller, S. Y. Erofeeva and P. M. Kosro 2003 The M₂ internal tide off Oregon: inferences from data assimilation *J. Phys. Ocean.* **33** 1733-1757
- Haber, E., Ascher, U. M., Aruliah, D. A., and Oldenburg, D. W. 2000 On Optimisation Techniques for Solving Nonlinear Inverse Problems *Inverse Problems* **16** 1263-1280
- Haber, E. and U. Ascher 2001 Preconditioned all-at-once methods for large, sparse parameter estimation problems, *Inverse Problems* **17** 1847-1864
- Haber, E. 2005 Quasi-Newton Methods for Large-Scale Electromagnetic Inverse Problems *Inverse Problems* **21** 305-323
- Mackie, R.L and T.R. Madden 1993 Conjugate direction relaxation solutions for 3-D magnetotelluric modeling *Geophysics* **58** 1052-1057
- Marquardt, D. W. 1963 An Algorithm for least-squares estimation of nonlinear parameters *J. Soc. Indust. Appl. Math* **11** 431-441
- Newman, G. A., and Alumbaugh, D. L. 1996 Three-dimensional massively parallel electromagnetic inversion-I. Theory *Geophys. J. Int.* **128** 340-354
- Newman, G. A., and Alumbaugh, D. L. 2000 Three-Dimensional Magnetotelluric Inversion Using Non-Linear Conjugate Gradients *Geophys. J. Int.* **140** 410-424
- Newman, G. A., and Boggs, P. T. 2004 Solution Accelerators for Large-Scale Three- Dimensional Electromagnetic Inverse Problem *Inverse Problems* **20** s151-s170
- Nocedal, J., and Wright, S. 1999 *Numerical Optimization* (New York: Springer-Verlag)

- Oldenburg, D. W., McGillivray, P. R., and Ellis, R. G. 1993 Generalized subspace methods for large scale inverse problems *Geophys. J. Int.* **114** 12-20
- Parker, R.L. 1994 *Geophysical Inverse Theory*, (Princeton NJ: Princeton University Press)
- Parker, R. L., and L. Shure 1982 Efficient modelling of the Earth's magnetic field with harmonic splines *Geophys. Res. Lett.* **9** 812-815
- Parker, R.L, L. Shure, and J. A., Hildebrand 1987 The application of inverse theory to seamount magnetism *Rev. Geophys.* **25** 17-40
- Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T. 1986 *Numerical Recipes: The Art of Scientific Computing* (Cambridge: Cambridge University Press)
- Rodi, W. L. 1976 A technique for improving the accuracy of finite element solutions for magnetotelluric data *Geophys. J. R. Astr. Soc.* **44** 483-506
- Rodi, W. L., and Mackie, R. L. 2001 Nonlinear Conjugate Gradients Algorithm for 2-D Magnetotelluric Inversion *Geophysics* **66** 174-187
- Sasaki, Y. 2004 Three-dimensional inversion of static-shifted magnetotelluric data *Earth, Planets, Space* **56** 239-248
- Schultz, A., Kurtz, R.D., Chave, A.D., and Jones, A.G. 1993 Conductivity discontinuities beneath a stable craton *Geophys. Res. Lett.* **24** 2941-2944
- Siripunvaraporn, W. and G. Egbert 2000 An Efficient Data-Subspace Inversion for Two-Dimensional Magnetotelluric Data *Geophysics* **65** 791-803
- Siripunvaraporn, W., G. Egbert, and Y. Lenbury 2002 Numerical accuracy of magnetotelluric modeling: A comparison of finite difference approximations *Earth, Planets, Space* **54** 721-726
- Siripunvaraporn, W., M. Uyeshima and Egbert, G. 2004 Three-dimensional inversion for Network-Magnetotelluric data *Earth, Planets, Space* **56** 893-902
- Siripunvaraporn, W., Egbert, G., Lenbury, Y., and Uyeshima, M. 2005 Three-Dimensional Magnetotelluric Inversion: Data Space Method *Phys. Earth Planet. Inter.* **150** 3-14
- Siripunvaraporn W. and G. Egbert 2006 Data space conjugate gradient inversion for 2-D Magnetotelluric data submitted to *GJI*
- Smith, J. T. 1996 Conservative Modeling of 3-D Electromagnetic Fields, Part I, Properties and Error Analysis *Geophysics* **61** 1308-1318
- Smith J. T, and Booker, J. R. 1988 Magnetotelluric inversion for minimum structure *Geophysics* **53** 1565-1576
- Torres-Verdin, C., A. Malinverno, V.L. Druskin, S. Fang, and L.A. Knizhnerman 2000 A dual-grid nonlinear inversion technique with applications to the interpretation of dc resistivity data *Geophysics* **65** 1733-1745
- Uchida, T, 1993 Smooth 2-D Inversion for magnetotelluric data based on statistical criterion ABIC *Jour. Geomag. Geoelec.* **45** 841-858
- Xie, G., M.L. Oristaglio, J. Li, E.L. Majer, and D. Zuo 2000 3-D electromagnetic modeling and nonlinear inversion *Geophysics* **65** 804-822
- Yee, K. S. 1966 Numerical Solution of Initial Boundary Value Problems Involving Maxwells Equations in Isotropic Media *IEEE Trans. Ant. Prop.* **AP-14** 302-307
- Zhdanov, M.S., S. Fang, and G. Hursan 2000 Electromagnetic inversion using quasi-linear approximation *Geophysics* **65** 1501-1513
- Zhdanov, M. S., and E. Tartaras 2002 Inversion of multi-transmitter 3D electromagnetic data based on the localized quasi-linear approximation *Geophys J. Int.* **148** 506-519
- Zhang, J., R.L. Mackie, and T.R. Madden 1995 Three-dimensional resistivity forward modeling and inversion using conjugate gradients *Geophysics* **60** 1313-1325

Figure 1. Pseudo-code for (a) DASOCC, and (b) the hybrid scheme, for the simplified case $\mathbf{C}_m = \mathbf{I}$. For both schemes phase I of the Occam process is depicted; in phase II minimization of misfit is replaced by minimization of the model norm, subject to the constraint that the data are fit to the prescribed tolerance.

$$\mathbf{m}_0 = \text{prior model parameter}; \quad \mathbf{d} = (\mathbf{d}_1^T \mathbf{d}_2^T \dots \mathbf{d}_{N_f}^T) = \text{data}$$

(a) DASOCC algorithm

For $n = 0, 1, \dots$

{ Compute Jacobian: $\mathbf{J}^n = \partial \mathbf{f} / \partial \mathbf{m} |_{\mathbf{m}_n}$

$$\hat{\mathbf{d}} = \mathbf{d} - \mathbf{f}(\mathbf{m}_n) + \mathbf{J}^n(\mathbf{m}_n - \mathbf{m}_0)$$

Minimize over ν :

$$\mathbf{m}_{n+1} = \min! \|\mathbf{d} - \mathbf{f}(\mathbf{m}_\nu)\|^2 \text{ where } \mathbf{m}_\nu = \mathbf{J}(\mathbf{J}^n \mathbf{J}^{nT} + \nu^{-1} \mathbf{I})^{-1} \hat{\mathbf{d}} + \mathbf{m}_0$$

}

(b) Hybrid CG-DASOCC algorithm

Outer loop: for $n = 0, 1, \dots$

{ $\hat{\mathbf{d}} = \mathbf{d} - \mathbf{f}(\mathbf{m}_n) + \mathbf{J}^n(\mathbf{m}_n - \mathbf{m}_0)$

Frequency loop : for $l = 1, N_f$

{ Solve: $\mathbf{J}_l^n \mathbf{J}_l^n \mathbf{m} = \mathbf{J}_l^n \hat{\mathbf{d}}_l$ using CG:

$$\mathbf{r}_{1l} = \mathbf{g}_{1l} = -\hat{\mathbf{d}}_l \quad \mathbf{s}_{1l} = \mathbf{h}_{1l} = \mathbf{J}_l^{nT} \mathbf{r}_{1l}$$

CG Loop : for $k = 1, K - 1$

$$\{\mathbf{r}_{k+1,l} = \mathbf{J}_l^n \mathbf{h}_{kl}$$

$$\mathbf{s}_{k+1,l} = \mathbf{A} \mathbf{h}_{kl} = \mathbf{J}_l^{nT} \mathbf{J}_l^n \mathbf{h}_{kl} = \mathbf{J}_l^{nT} \mathbf{r}_{k+1,l}$$

$$\lambda_{kl} = \mathbf{h}_{kl}^T \mathbf{g}_{kl} / \mathbf{h}_{kl}^T \mathbf{A} \mathbf{h}_{kl} = \mathbf{h}_{kl}^T \mathbf{g}_{kl} / \mathbf{h}_{kl}^T \mathbf{s}_{k+1,l}$$

$$\mathbf{g}_{k+1,l} = \mathbf{g}_{kl} - \lambda_{kl} \mathbf{A} \mathbf{h}_{kl} = \mathbf{g}_{kl} - \lambda_{kl} \mathbf{s}_{k+1,l}$$

$$\gamma_{kl} = \mathbf{g}_{k+1,l}^T \mathbf{A} \mathbf{h}_{kl} / \mathbf{h}_{kl}^T \mathbf{A} \mathbf{h}_{kl} = \mathbf{g}_{k+1,l}^T \mathbf{s}_{k+1,l} / \mathbf{h}_{kl}^T \mathbf{s}_{k+1,l}$$

$$\mathbf{h}_{k+1,l} = \mathbf{g}_{k+1,l} - \gamma_{kl} \mathbf{h}_{kl}$$

Save: $\mathbf{r}_{k+1,l}, \mathbf{s}_{k+1,l}$

}

Orthonormalize:

$$\mathbf{U}_{Kl} = \mathbf{R}_{Kl} (\mathbf{R}_{Kl}^T \mathbf{R}_{Kl})^{-\frac{1}{2}} \quad \text{where } \mathbf{R}_{Kl} = (\mathbf{r}_{1l} \dots \mathbf{r}_{Kl})$$

$$\tilde{\mathbf{J}}_{Kl}^T = (\mathbf{s}_{1l} \dots \mathbf{s}_{Kl}) (\mathbf{R}_{Kl}^T \mathbf{R}_{Kl})^{-\frac{1}{2}}$$

}

$$\text{Set: } \tilde{\mathbf{J}}_K = (\tilde{\mathbf{J}}_{K1}^T \dots \tilde{\mathbf{J}}_{KN_f}^T) \quad \mathbf{U}_K = (\mathbf{U}_{K1}^T \dots \mathbf{U}_{KN_f}^T)$$

minimize over ν :

$$\mathbf{m}_{n+1} = \min! \|\mathbf{d} - \mathbf{f}(\mathbf{m}_\nu)\|^2 \text{ where}$$

$$\mathbf{m}_\nu = \tilde{\mathbf{J}}_K (\tilde{\mathbf{J}}_K \tilde{\mathbf{J}}_K^T + \nu^{-1} \mathbf{I})^{-1} \mathbf{U}_K^T \hat{\mathbf{d}} + \mathbf{m}_0$$

}

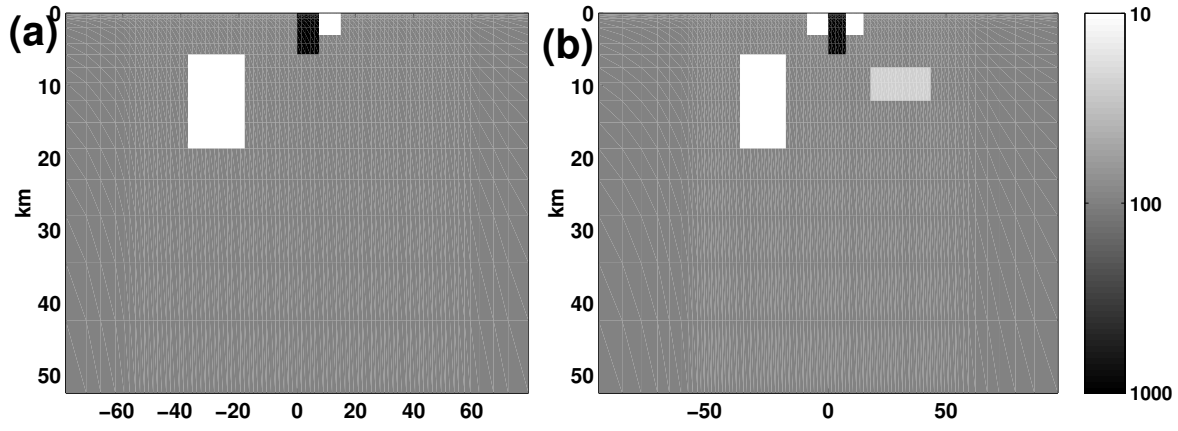


Figure 2. Models used in synthetic data inversion tests

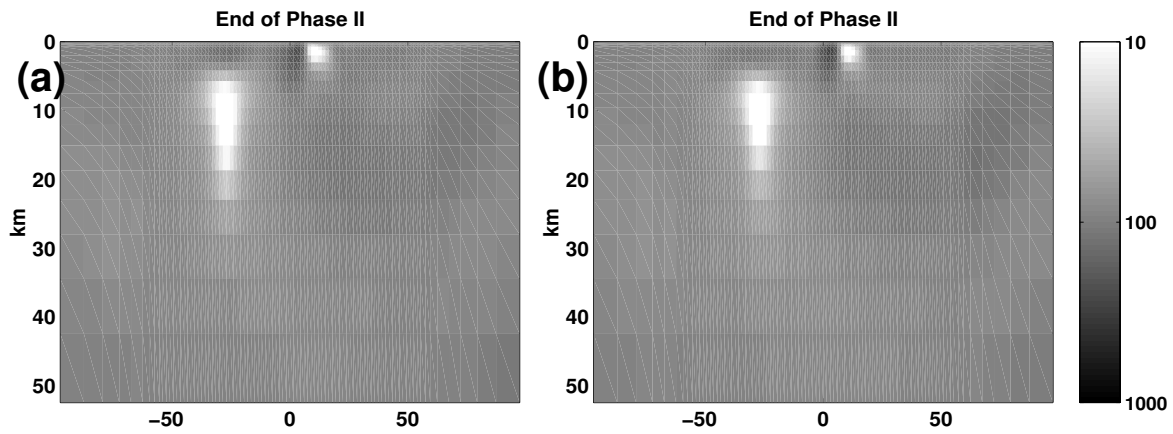


Figure 3. Inversion results for synthetic data from the simpler model of Figure 1a, obtained using (a) the hybrid CG-DASOCC scheme of section 3 and (b) A standard data space Occam scheme, based on calculation of the full Jacobian.

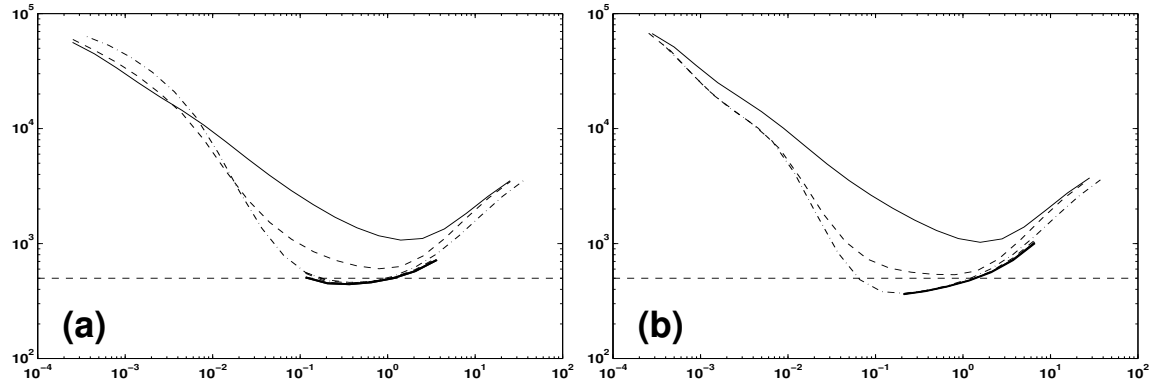


Figure 4. Total squared misfit as a function of μ for synthetic data set one. Squared misfit curves are plotted for a series of X outer loop iterations for (a) the hybrid CG-DASOCC scheme of section 3 and (b) A standard data space Occam scheme, based on calculation of the full Jacobian.

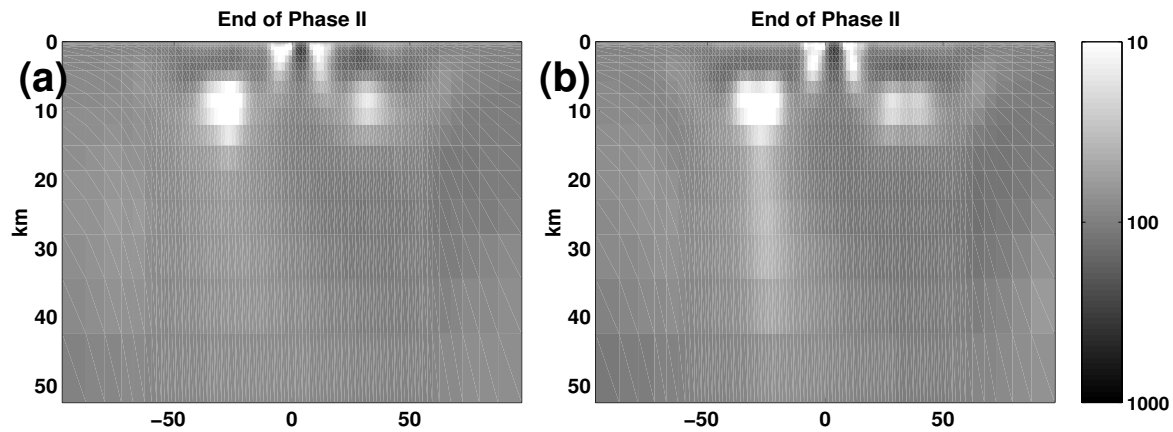


Figure 5. Inversion results for synthetic data from the model of Figure 1b, obtained using (a) the hybrid CG-DASOCC scheme of section 3 and (b) A standard data space Occam scheme, based on calculation of the full Jacobian.

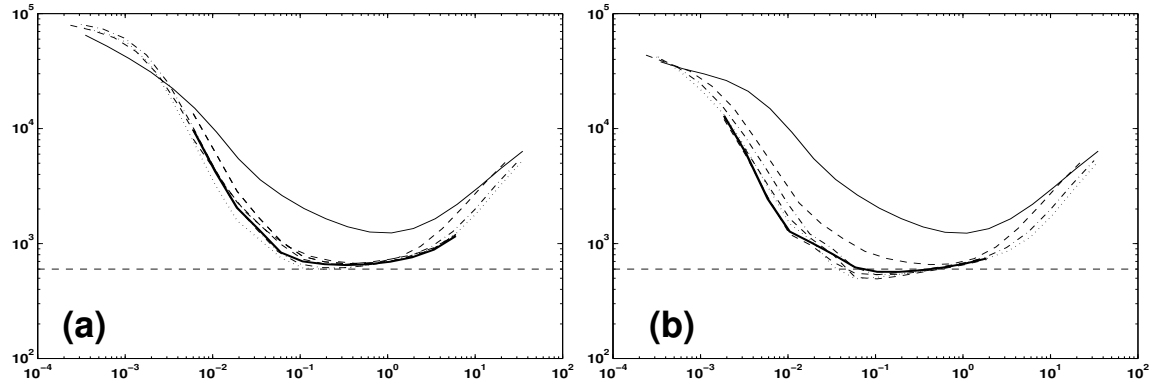


Figure 6. Total squared misfit as a function of μ for synthetic data set two. Squared misfit curves are plotted for a series of X outer loop iterations for (a) the hybrid CG-DASOCC scheme of section 3 and (b) A standard data space Occam scheme, based on calculation of the full Jacobian.